

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica



Desarrollo del Sistema de Mensajería de Texto para Telefonía Fija

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en
Electrónica con el grado académico de Licenciatura**

Eduardo Alfaro Hernández

Victor Mendivil Piedra

Cartago, Febrero de 2006

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

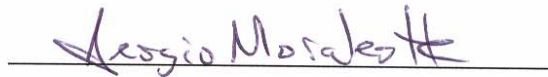
Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



MSc. Gabriela Ortiz León

Profesora lectora



Ing. Sergio Morales Hernández

Profesor lector



Ing. Faustino Montes de Oca

Profesor asesor



Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 2 de Febrero de 2006

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 2 febrero 2006



Eduardo Alfaro Hernández

Céd: 1-1120-0986

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 2 febrero 2006


Victor Mendivil Piedra

Céd: 1-1096-0252

Resumen

El Instituto Costarricense de Electricidad (ICE) tiene como ideal la innovación tecnológica y búsqueda de nuevos servicios, lo cual crea la necesidad de ofrecer y ampliar los servicios de comunicación entre sus clientes. La comunicación telefónica se divide en la telefonía celular y telefonía fija. Actualmente, la principal fuente de ingresos del ICE se da por medio de los servicios en la red fija. La telefonía celular está tomando preferencia entre los clientes, es por ello que se debe ofrecer nuevos servicios en telefonía fija.

El objetivo del proyecto es desarrollar un nuevo servicio para los clientes de la red fija, éste consiste en el envío de mensajes cortos de texto a través de la línea fija. Para lograr la estandarización del servicio, en el desarrollo del sistema, se tomaron como referencia los protocolos y normas definidos por el Instituto de Estandarización de Telecomunicaciones Europeo (ETSI).

La importancia de utilizar un estándar internacional radica en la posibilidad de utilizar terminales comerciales (teléfonos) diseñadas para el envío de mensajes cortos de texto. De este modo, el proyecto se enfoca en desarrollar el centro de mensajes (SM-SC) y crear la aplicación que maneje el protocolo.

El centro de mensajes se desarrolló en una la plataforma de valor agregado, marca APEX, en ambiente Windows 2000. Esta plataforma se conecta a la red telefónica nacional (PSTN) a través de un canal E1 con señalización ISDN.

Una de las metas del ICE, en un futuro, es lograr la integración de los diferentes servicios que brinda en todas las tecnologías (telefonía móvil y fija). Con este proyecto se permite el envío de mensajes de texto desde y hacia teléfonos fijos, la utilización del estándar deja la puerta abierta para poder desarrollar el intercambio de mensajes de texto con la telefonía celular.

Palabras claves: SMS línea fija, teléfonos SMS

Abstract

The technological innovation and search of new services is an ideal of the Instituto Costarricense de Electricidad (ICE), which makes the necessity to offer and to enlarge the communication services among its clients. The phone communication is divided in the cellular telephony and fixed telephony. At the moment, the main income of ICE it is given by means of the fixed telephony services. The cellular telephony is taking preference among the clients, therefore it's necessary to offer new services for fixed telephony.

The objective of the project is to develop a new service for the clients of the fixed telephony network, which consists in providing the short text message service (SMS) for fixed lines. In order to standardize the service, in the development of the system, protocols and norms defined by the European Telecommunications Standards Institute (ETSI) have been taken as a reference.

The importance of using an international standard resides in the possibility of using commercial terminals (telephones) designed for the SMS. This way, the project is focused in developing the center of messages (SM-SC) and to create the application that manages the protocol.

The center of messages was developed in an APEX platform of added value, under a Windows 2000 environment. This platform is connected to the Public Switched Telephone Network (PSTN) through a channel E1 with signaling ISDN.

In the future, one of ICE's goal is to allow the integration of the different services in all technologies (mobile and fixed telephony). With this project the SMS is allowed between fixed telephones, the use of the standard leaves a door open to be able to develop the exchange of text messages with the cellular telephony.

Key words: SMS for fixed lines, SMS phones, fixed line telephony short message service

ÍNDICE GENERAL

Capítulo 1: Introducción	10
1.1 Problema existente e importancia de su solución	10
1.2 Solución seleccionada	11
Capítulo 2: Meta y Objetivos	13
2.1 Meta.....	13
2.2 Objetivo General.....	13
2.3 Objetivos Específicos.....	13
2.3.1 Objetivos de Capa Física	13
2.3.2 Objetivos de Capa de Enlace de Datos.....	14
2.3.3 Objetivos de Capa de Transferencia	14
2.3.4 Objetivos de Capa de Aplicación.....	15
2.4 Objetivos de documentación.....	15
2.5 Objetivos de implementación	15
Capítulo 3: Marco Teórico	16
3.1 Descripción general del sistema de mensajería de texto para telefonía fija	16
3.2 Antecedentes Bibliográficos.....	18
3.3 Plataforma para el desarrollo del centro de mensajes	21
Capítulo 4: Procedimiento Metodológico.....	23
4.1 Reconocimiento y definición del problema.....	23
4.2 Obtención y análisis de información	23
4.3 Evaluación de las alternativas y síntesis de una solución.....	23
4.4 Implementación de la solución.....	24
4.5 Reevaluación y rediseño.....	24
Capítulo 5: Descripción detallada de la solución	25
5.1 Análisis de soluciones y selección final	25
5.2 Descripción del hardware	26
5.3 Descripción del software.....	27
Capítulo 6: Análisis de Resultados.....	48
Capítulo 7: Conclusiones y recomendaciones	60
7.1 Conclusiones	60
7.2 Recomendaciones	61
Bibliografía	62
Apéndices	64
A.1 Abreviaturas	64
A.2 Información sobre la institución	66
A.3 Procesos del Protocolo 1.....	67
A.4 Arquitectura del protocolo 1	70

ÍNDICE DE FIGURAS

Figura 1.1	Situación actual de la red telefónica	10
Figura 1.2.1	Sistema general.....	11
Figura 3.1.1	Diagrama de bloques del servicio de mensajería de texto para telefonía fija	16
Figura 3.1.2	Principio de transferencia de mensajes cortos	17
Figura 3.2.1.1	Trama de la capa de enlace de datos del Protocolo 1.....	19
Figura 3.2.1.2	Trama de la capa de enlace de datos del Protocolo 2.....	19
Figura 3.3.1	Componentes del sistema OmniVox AES	22
Figura 5.1.1	Diagrama de bloques del sistema de mensajería en telefonía fija.	25
Figura 5.3.1.1	Flujo de la aplicación para la recepción de SM	28
Figura 5.3.1.2	Configuración del icono <i>Start</i>	28
Figura 5.3.1.3	Configuración del icono <i>Call Info</i>	29
Figura 5.3.1.4	Configuración de la función <i>Accept Call</i>	29
Figura 5.3.1.5	Configuración del icono <i>Driver SMS</i>	31
Figura 5.3.1.6	Diagrama de flujo de la clase “Manejo de Recurso de Voz”	32
Figura 5.3.1.7	Diagrama de flujo del procedimiento “Capa Enlace Datos”	34
Figura 5.3.1.8	Estructura para la trama error.....	35
Figura 5.3.1.9	Diagrama de flujo de la clase Capa de Transferencia	36
Figura 5.3.2.1	Flujo de la llamada para la entrega de mensajes de texto	37
Figura 5.3.2.2	Configuración del <i>Start</i>	37
Figura 5.3.2.3	Definición de variables	38
Figura 5.3.2.4	Configuración de <i>Chequear BD</i>	39

Figura 5.3.2.5	Configuración de la llamada	40
Figura 5.3.2.6	Configuración de <i>Driver FSMS</i>	41
Figura 5.3.2.7	Diagrama de flujo de la capa de transferencia para las rutina de enviar mensaje	42
Figura 5.3.2.8	Paquete del <i>SMS-Deliver</i>	43
Figura 5.3.2.9	Paquete del <i>SMS-Submit</i>	43
Figura 5.3.2.10	Diagrama de flujo del codificador de 7 bits	45
Figura 5.3.2.11	Diagrama de flujo de la capa de enlace de datos, para las rutinas de envío de mensaje.....	46
Figura 6.1.1	Medición de frecuencia para los bits de marca (“1” lógico)	48
Figura 6.1.2	Medición de frecuencia para los bits de espacio (“0” lógico)	49
Figura 6.2.1	Diagrama de flujo de un SMS-Submit en caso normal	50
Figura 6.2.2	Trama <i>DLL_SMS_EST</i> enviada por el SM-SC	51
Figura 6.2.3	Trama <i>DLL_SMS_DATA</i> enviada por el SM-TE	51
Figura 6.2.4	Trama <i>DLL_SMS_ACK</i> enviada por el SM-SC.....	53
Figura 6.2.5	Trama <i>DLL_SMS_REL</i> enviada por el SM-TE	53
Figura 6.3.1	Tramas en el proceso de <i>Deliver</i> en caso normal	54
Figura 6.3.2	Trama del <i>Deliver</i>	55
Figura 6.3.3	PDU del <i>Deliver</i>	56
Figura 6.4.1	Ejemplo de un mensaje guardado en la base de datos.....	57

ÍNDICE DE TABLAS

Tabla 5.3.2.1	Representación del SCTS.....	44
Tabla 6.2.1	Tramas recibidas y enviadas de un <i>SMS_SUBMIT</i>	50
Tabla 6.2.2	Decodificación del PDU recibido en el <i>DLL_SMS_DATA</i>	52
Tabla 6.2.3	Decodificación del PDU recibido en el <i>DLL_SMS_ACK</i>	53
Tabla 6.3.1	Resultados de las tramas del proceso de <i>Deliver</i>	55

Capítulo 1: Introducción

En este capítulo se expone la problemática existente que motivó al Instituto Costarricense de Electricidad (ICE) para impulsar el desarrollo del proyecto “Desarrollo del Sistema de Mensajería de Texto para Telefonía Fija”, además se explica la selección de la solución.

1.1 Problema existente e importancia de su solución

La red telefónica nacional actual cuenta con servicios de telefonía fija y móvil, a través de éstos es posible comunicarse por medio de la voz en cualquier dirección posible, es decir entre telefonía fija a fija, de fija a móvil, de móvil a fija y de móvil a móvil. Otro servicio que está tomando un gran mercado es el envío de mensajes cortos de texto (SMS). Actualmente este servicio sólo se brinda para telefonía móvil, de aquí nace la necesidad de extender esta opción de comunicación para la telefonía fija. En la siguiente figura se ilustra esta situación.

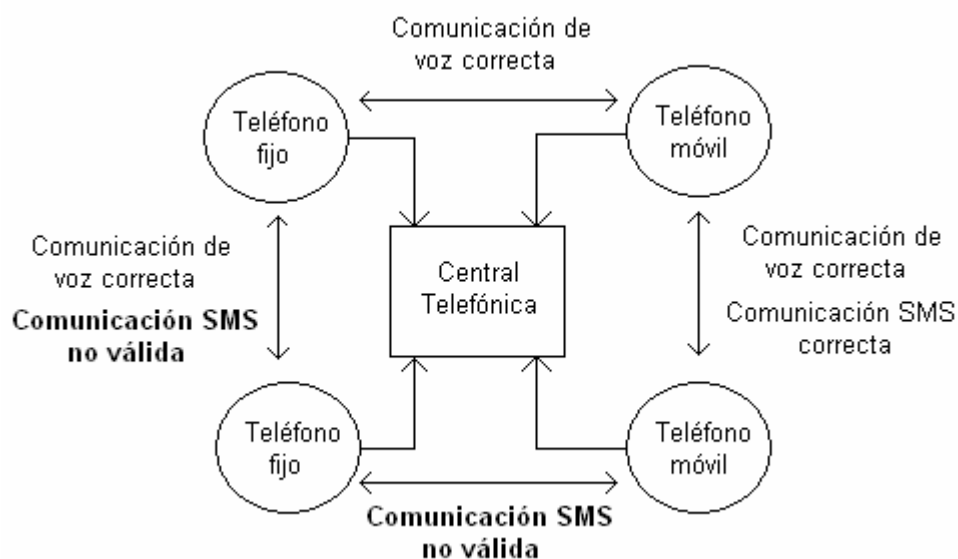


Figura 1.1 Situación actual de la red telefónica

Como se observa en la figura, existen dos rutas sobre las cuales no es posible el envío de mensajes cortos de texto, en ambas se debe manejar aspectos de hardware y software para llegar a obtener los enlaces. El enfoque en este

proyecto es lograr un mecanismo que soporte el envío de mensajes de texto entre teléfonos fijos.

El envío de mensajes de texto por medio de teléfonos móviles es posible debido a que las centrales (para telefonía celular) ya tienen los sistemas de identificación adecuados, y la central se encarga de re-direccionar los mensajes hacia un centro de operación de mensajería corta (SM-SC). Para la telefonía fija no se cuenta con esta facilidad, pues ni las terminales (teléfonos fijos) ni las propias centrales tienen actualmente un sistema que soporte el envío de mensajería corta.

Entre las principales razones para desarrollar el proyecto está la posibilidad de que los clientes del ICE tengan una nueva forma para comunicarse en telefonía fija, y satisfacer la necesidad de comunicación en los usuarios del ICE que tengan discapacidad de hablar u oír. A su vez, se intenta generar más ingresos para la institución.

1.2 Solución seleccionada

El servicio de mensajería de texto en telefonía fija en el territorio nacional es algo nuevo, y no se tenía el conocimiento de cómo debía funcionar, por lo tanto, se realizó un estudio en países donde actualmente se brinda este servicio, países europeos al igual que en Estados Unidos. Con base en el estudio, se determinó que el sistema para el servicio de mensajería de texto en telefonía fija se conforma por 3 bloques: la terminal del usuario (SM-TE), la red de telefonía nacional (PSTN) y el centro de mensajes (SM-SC). Esto se ilustra en la figura 1.2.1.

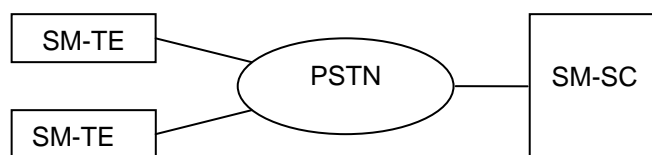


Figura 1.2.1 Sistema general

La terminal del usuario es un teléfono especial que soporta la mensajería de texto para telefonía fija (tiene un teclado alfanumérico y una pantalla). La red telefónica es el medio por donde se transmite información a través de una llamada normal de voz y el centro de mensajes es el encargado de recibir los mensajes de texto desde una terminal de origen y enviarlos hacia una terminal de destino.

Ya que las terminales del usuario existen comercialmente, el proyecto consistió en desarrollar el centro de mensajes. Para que fuera funcional el centro de mensajería se buscó estándares internacionales. Al inicio se tenían dos estándares: el europeo definido por el Instituto de Estandarización de Telecomunicaciones Europeo (ETSI) y el estándar americano ADSI (*Analog Display Services Interface*) de BellCore utilizado en Estados Unidos.

Para determinar cual de los dos estándares utilizar (el europeo o el americano) se consultó con funcionarios del ICE, se determinó que el equipo actual del ICE utiliza estándares europeos. Por este motivo, se seleccionó el estándar de la ETSI. Este estándar especifica la posibilidad de emplear dos protocolos: Protocolo 1 y Protocolo 2.

La principal diferencia entre estos dos protocolos es que el Protocolo 1 está orientado a hacia telefonía móvil (tecnología GSM), mientras que el Protocolo 2 está orientado exclusivamente a telefonía fija. Como en un futuro se pretende enlazar los servicios de mensajería de telefonía fija y móvil, se escogió el Protocolo 1.

Una vez que se estableció el protocolo ha utilizar, se pudo definir sobre cual equipo se iba ha implementar el centro de mensajería, para esto se seleccionó una plataforma de valor agregado de marca APEX, debido a que permite la conexión con la red telefónica (realizando la función de puerta de entrada del SM-SC), además permite implementar el protocolo utilizando la herramienta de diseño de Visual C++.

Capítulo 2: Meta y Objetivos

2.1 Meta

Ampliar la variedad de servicios de comunicación que el ICE ofrece a los clientes de la red fija, mediante el desarrollo del servicio de mensajería de texto para las líneas fijas.

2.2 Objetivo General

Desarrollar el centro de mensajería de texto para telefonía fija que cumpla con el estándar de la ETSI. Este centro de mensajería incluye el manejo del protocolo 1 y además la aplicación general que administre el servicio.

2.3 Objetivos Específicos

2.3.1 Objetivos de Capa Física

1. Desarrollar la rutina en el centro de mensajes que realice la inicialización de las tarjetas *Dialogic*, de modo que permita abrir los canales y enlazar las tarjetas de recursos de voz con la de los recursos de red.
2. Desarrollar la rutina en el centro de mensajes que configure la tarjeta de voz para que pueda recibir y enviar información modulada en FSK, según como lo define el protocolo 1 de la ETSI.
3. Desarrollar la rutina en el centro de mensajes que permita recibir y almacenar la información, para que pueda ser utilizada por la capa de enlace de datos.
4. Desarrollar la rutina en el centro de mensajes que permita transmitir la información de la capa de enlace hacia la red telefónica (PSTN).
5. Desarrollar la rutina en el centro de mensajes que libere los recursos de las tarjetas de red y voz, al cerrar los canales utilizados en el envío y recepción de datos.

2.3.2 Objetivos de Capa de Enlace de Datos

1. Desarrollar la rutina del centro de mensajes que permita verificar la validez de las tramas recibidas en la capa de enlace de datos (Capa 2).
2. Desarrollar la rutina que le permita al centro de mensajes armar y enviar tramas con los diferentes tipos de mensajes (mensajes de: error, datos, liberación, reconocimiento o no reconocimiento).
3. Desarrollar la rutina dentro del centro de mensajes que pueda calcular el valor del *checksum* para tramas que se envían, así como para chequear tramas que se reciben (provenientes de la terminal del cliente).
4. Desarrollar la rutina dentro del centro de mensajes que permita dividir mensajes con longitudes mayores que 176 octetos, en dos o más paquetes y controlar el envío de cada uno de estos.
5. Desarrollar un temporizador dentro del centro de mensajes que permita el control del tiempo de respuesta en el intercambio de información entre el centro de mensajes y la terminal.

2.3.3 Objetivos de Capa de Transferencia

1. Desarrollar la rutina del centro de mensajes que permita codificar y decodificar el mensaje de texto, ya sea en 7 bits o 8 bits.
2. Desarrollar rutinas en el centro de mensaje que permitan manipular los paquetes a nivel de byte y de bits. Y de este modo, poder configurar los parámetros de los paquetes según el protocolo 1.
3. Desarrollar rutinas que permitan la lectura y escritura en bases de datos para obtener información de los mensajes de texto y generar reportes en el centro de mensajes.
4. Desarrollar la rutina que verifique la validez del paquete recibido, y que éste sea soportado por el centro de mensajes, revisando para ello, el esquema de codificación y el identificador de protocolo.

5. Desarrollar la rutina que arme el paquete (capa 3) para el *Submit-Report* (confirmación del mensaje recibido por el SMSC) y *Deliver* (mensaje enviado por el SMSC).

2.3.4 Objetivos de Capa de Aplicación

1. Desarrollar mediante el software OmniView, la aplicación del centro de mensajes que administre de forma general el servicio de mensajería de texto.

2.4 Objetivos de documentación

1. Crear un documento con las especificaciones, restricciones y recomendaciones del centro de mensajes desarrollado.

2.5 Objetivos de implementación

1. Desarrollar el prototipo del centro de mensajes.

Capítulo 3: Marco Teórico

En este capítulo se hará referencia a la descripción general del servicio de mensajería de texto para telefonía fija, así como al protocolo que se emplea (protocolo 1) y sus capas. Además, se hace una descripción del funcionamiento del sistema telefónico nacional.

3.1 Descripción general del sistema de mensajería de texto para telefonía fija

Para implementar el servicio de mensajería de texto en teléfonos fijos se requiere de un centro de mensajes (SM-SC), el cual debe recibir el mensaje y luego reenviarlo al usuario destino (similar al servicio de mensajes de texto en teléfonos móviles).

En la figura 3.1.1 se muestra la arquitectura del servicio de mensajería corta (SMS) para el sistema de telefonía fija (PSTN). El servicio está constituido por tres bloques: la terminal de mensajería corta del usuario (SM-TE), el centro de mensajería corta (SM-SC) y la red o sistema de telefonía fija.

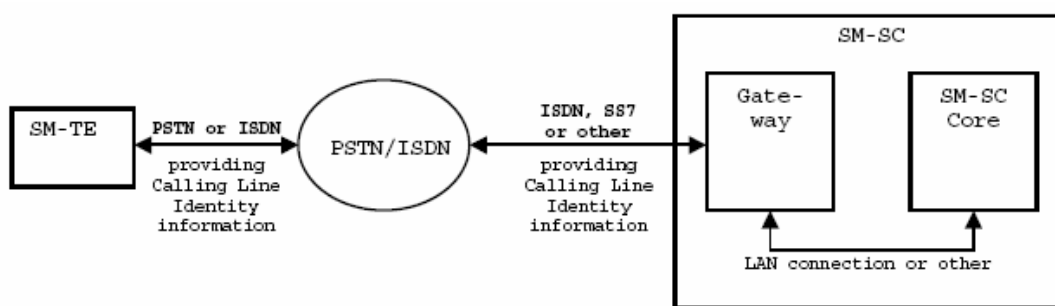


Figura 3.1.1 Diagrama de bloques del servicio de mensajería de texto para telefonía fija ¹

¹ Tomado de ETSI ES 201 912 V1.1.1 (2002-01)

La terminal del usuario deberá estar conectada a la red de telefonía fija y el centro de mensajería corta accederá a la red a través de un enlace primario que maneje el protocolo de señalización 7 (SS7) o ISDN.

Para enviar y recibir mensajes cortos, se usa el ancho de banda de voz que se establece en la red de telefonía fija entre la terminal del usuario y el centro de mensajes. El principio general para el envío y recepción de mensajes cortos se ilustra en la figura 3.1.2.

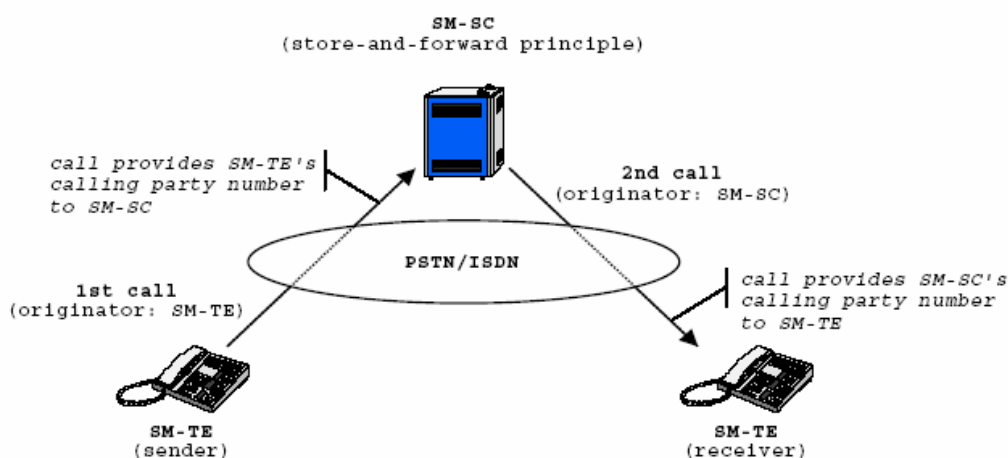


Figura 3.1.2 Principio de transferencia de mensajes cortos¹

La transferencia de mensajes cortos se puede describir en dos pasos: el envío del mensaje corto (transferencia del mensaje de la SM-TE hacia el SM-SC) y la recepción del mensaje corto (transferencia del mensaje del SM-SC hacia la SM-TE).

En el primer paso (envío del mensaje), la SM-TE realiza una llamada al SM-SC; como se ve en la figura 3.1.2, la red de telefonía fija deberá proveer el número de la terminal (servicio de identificación de llamadas: *Caller ID*), con el fin de que el centro de mensajes pueda identificar el número telefónico de la terminal que envía el mensaje. Además, esta información puede ser usada para propósitos de facturación.

¹ Tomado de ETSI ES 201 912 V1.1.1 (2002-01)

Después de que se establece la conexión (ancho de banda de voz) entre la SM-TE y el SM-SC, se inicia la fase de transferencia de la información de la terminal hacia el centro de mensajes. Una vez que la transferencia se haya completado, la conexión entre ambas entidades es liberada.

En el segundo paso (recepción del mensaje), el SM-SC realiza una llamada a la SM-TE destino para entregar el mensaje. En este caso, la red de telefonía deberá proveer el número telefónico del SM-SC a la SM-TE. La terminal usa esta información para identificar y conectarse automáticamente o realizar la llamada al SM-SC. Al igual que en el primer paso, el mensaje es transmitido después de que la conexión entre las dos entidades se establece, pero ahora desde el SM-SC hasta el SM-TE.

3.2 Antecedentes Bibliográficos

Con la investigación realizada se encontraron dos estándares internacionales que permiten el servicio de mensajería de texto para telefonía fija; el primero de ellos es el estándar de la ETSI (europeo) y el segundo es el estándar ADSI de BellCore (Estados Unidos). A continuación se explican las características generales de los dos estándares.

3.2.1 Características del estándar ETSI

El estándar europeo puede usar dos protocolos: el Protocolo 1 y el Protocolo 2. Ambos protocolos tienen tres capas, estas capas son:

- Capa física
- Capa de enlace de datos
- Capa de transferencia

La capa física permite enviar y recibir información utilizando el ancho de banda del canal de voz. Toda comunicación es *half duplex*, por medio de modulación FSK dentro del ancho de banda del canal de voz (300 a 3400Hz) y con velocidad de 1200 baudios.

Las entidades (SM-SC y SM-TE) deberán ser capaces de recibir y enviar señales de FSK con las siguientes frecuencias:

- Marca = "1" lógico, frecuencia: 1 300 Hz \pm 1,5 %.
- Espacio = "0" lógico, frecuencia: 2 100 Hz \pm 1,5 %.

En los dos protocolos, la capa de enlace de datos permite verificar la confiabilidad de las tramas recibidas mediante la detección de errores en las mismas. Aunque su función es la misma, los elementos de las tramas son diferentes, en las figuras 3.2.1.1 y 3.2.1.2 se muestran las composiciones de las tramas de capa de enlace de datos del Protocolo 1 y del Protocolo 2 respectivamente.

Bits de marca	Tipo de mensaje	Longitud mensaje	Mensaje	Check-sum
---------------	-----------------	------------------	---------	-----------

Figura 3.2.1.1 Trama de la capa de enlace de datos del Protocolo 1

Ancho de canal	Bits de marca	Tipo de mensaje	Longitud mensaje	Mensaje	Check-sum
----------------	---------------	-----------------	------------------	---------	-----------

Figura 3.2.1.2 Trama de la capa de enlace de datos del Protocolo 2

También es importante señalar que para el Protocolo 1 el valor de los bits de marca es de 80 bits, mientras que para el Protocolo 2 los bits de marca son 300 bits.

La capa de transferencia del Protocolo 1 es la misma que se utiliza en la mensajería de texto en telefonía móvil GSM. La capa de transferencia del protocolo 2 emplea un número variable de parámetros relacionados con el entorno de la red telefónica fija (por ejemplo: concerniente a aspectos de privacidad).

3.2.2 Características del estándar ADSI de BellCore

Éste es un estándar que define un protocolo usado para la transmisión de información en FSK. La transmisión se da a través de las líneas telefónicas fijas, permitiendo una comunicación asincrónica de 8 bits, 1 bit de inicio y 1 bits de parada y sin paridad.

El protocolo ADSI, al igual que los protocolos europeos, está compuesto por 3 capas bien definidas, las cuales son:

- Capa física
- Capa de enlace de datos
- Capa de ensamble del mensaje

En la capa física se envía la información bit por bit, utilizando el estándar de un modem de 1200 baudios a través del canal de voz. La capa de enlace de datos emplea el *checksum* para la verificación de errores en las tramas. La capa de ensamble del mensaje se encarga de armar el cuerpo del mensaje. Después de que la información es transmitida, el teléfono envía un mensaje de reconocimiento el cual corresponde a tonos de multifrecuencia (DTMF); este tipo de respuesta es una de las diferencias que existe con respecto al estándar europeo, ya que en éste siempre hay un intercambio de tramas para confirmar la información. También difieren en las características de la capa física, la configuración del ancho del canal y los bits de marca y los temporizadores para el intercambio de información.

3.2.3 Descripción del servicio British Telecom

British Telecom es una empresa que presta el servicio de mensajería de texto en telefonía fija en Europa, el cual está creciendo fuertemente, con un promedio de 48 millones de mensajes enviados por día y ha llegado a desplazar en gran medida al e-mail y las llamadas, debido a su gran facilidad de uso y al bajo costo del mensaje. Para la utilización del servicio de envío de mensajería de

texto se debe contar con un teléfono compatible con texto y una línea telefónica también compatible, además de suscribirse al servicio *BT Caller Display*.

La terminal para el SMS fijo es una unidad telefónica diseñada para:

- Escribir mensajes
- Guardar mensajes de entrada
- Guardar mensajes de salida
- Soportar el protocolo SMS para telefonía fija (ETSI)

La línea fija SMS ofrece los siguientes servicios:

- Enviar mensajes desde telefonía fija hacia otra terminal fija, de telefonía fija hacia un teléfono celular (tecnología GSM), de teléfono celular hacia una terminal fija.
- También permite enviar mensajes hacia fax, obtener información como el tiempo, el tipo de cambio, etc.

3.3 Plataforma para el desarrollo del centro de mensajes

El centro de mensajes se implementó en una plataforma APEX. Esta plataforma cuenta con el sistema *Onmivox APEX Enhanced Switch* (AES), el cual corre sobre Microsoft Windows 2000. Este software permite desarrollar aplicaciones telefónicas.

Onmivox AES está desarrollado para hardware basado en arquitectura Pentium de Intel sobre buses PCI ó cPCI y usa tarjetas *Dialogic* también de Intel para el procesamiento de voz y medios. Los sistemas pueden ser configurados para trabajar con redes de datos múltiples y sistemas de alta capacidad de respaldos.

Los componentes individuales de las aplicaciones pueden correr en uno o múltiples sistemas para mejorar el desempeño (equilibrar la carga del sistema) o requerimientos de redundancia.

Los diferentes componentes del sistema AES se muestran en la figura 3.3.1.

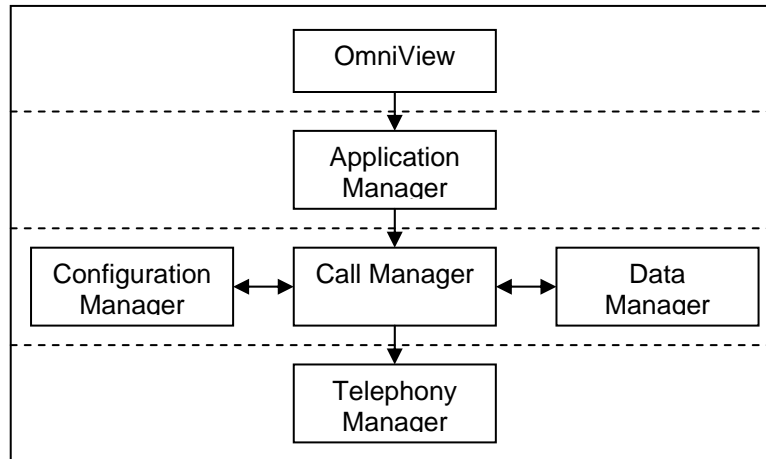


Figura 3.3.1 Componentes del sistema OmniVox AES

- *OmniView:*
Interface del usuario de Ominvox AES. Permite configuración, administración y creación de servicios (aplicaciones telefónicas).
- *Application Manager:*
Lleva la lógica y el control de las aplicaciones, corre aplicaciones creadas en Omniview.
- *Configuration Manager:*
Base de datos con la configuración central del OmniVox AES. Almacena: información de la configuración de todos los componentes de OmniVox, la lógica de las aplicaciones y archivos de voz.
- *Call Manager:*
Centro de control de AES. Se encarga de enrutar mensajes entre: Telephony Manager y Application Manager. Administra los recursos y el control de enrutamiento de las aplicaciones.
- *Data Manager:*
Almacena información acerca de eventos, alarmas, errores, registros detallados de las llamadas.
- *Telephony Manager:*
Servidor de telefonía de AES (parte física), depende de la aplicación y la configuración. PCI: +500 puertos, cPCI: +1000 puertos.

Capítulo 4: Procedimiento Metodológico

4.1 Reconocimiento y definición del problema

La identificación del problema se dio por parte de funcionarios del ICE, a raíz de la necesidad de ampliar los servicios que presta esta institución. La UEN Servicio al cliente (UEN-SC) es el departamento dentro del ICE que impulsó el desarrollo de este proyecto.

Para realizar el reconocimiento del problema se programaron entrevistas con proveedores y funcionarios del ICE de mucha experiencia en el área de las telecomunicaciones, con el fin de comprender aspectos relevantes de la red nacional de comunicaciones, tanto de la red telefonía celular como fija.

4.2 Obtención y análisis de información

Se obtuvo información relevante por medio de las entrevistas programadas y a partir de esta información se procedió a realizar una investigación bibliográfica por medio de libros suministrados por el ICE, revistas técnicas, Internet y consultando los estándares internacionales (ETSI).

Se analizó la información obtenida y se expuso la solución a los ingenieros del ICE para determinar el tipo de limitación que se podía dar; es decir, las limitación tecnológica en cuanto a la factibilidad de usar el equipo propiedad del ICE.

4.3 Evaluación de las alternativas y síntesis de una solución

El planteamiento de la solución estuvo sujeto a tres puntos claves: entrevistas, información bibliográfica y al equipo actual del ICE. El procedimiento seguido en la evaluación y discriminación de las diferentes propuestas es el siguiente:

- Si la solución es aplicable sólo a un tipo de central o se puede extender sin importar el fabricante de la central, es decir, que la

solución se pueda aplicar independientemente a una central Alcatel, Lucent, Nortel, Ericsson, Siemens.

- Factibilidad tecnológica, si la plataforma cuenta con el hardware y software necesarios para la solución.
- Que no sea una solución de final cerrado, que a través de la misma solución se puedan implementar otros servicios en el futuro.
- Que a través a la solución se logre una estandarización del servicio.

4.4 Implementación de la solución

Para el desarrollo del centro de mensajes se siguió una serie de pasos que se describen a continuación:

- Estudio del protocolo 1 de las normas ETSI (es_201912v010201p y el ts_100901v070500p).
- Capacitación para el manejo de la plataforma.
- Desarrollo de rutinas de prueba para el manejo de la plataforma.
- Desarrollo de las rutinas de la capa de enlace de datos y conexión con la capa física.
- Desarrollo de las rutinas de la capa de transferencia y conexión con la capa de enlace de datos.
- Desarrollo de la aplicación final para el servicio de mensajería de texto.

4.5 Reevaluación y rediseño

Debido a que el problema se trabajó de forma modular, es posible modificar independientemente cada módulo, siempre y cuando se respete los parámetros de entradas y salidas que tiene cada uno. Por lo tanto, cada módulo debió ser evaluado a través de rutinas de prueba para comprobar su correcto funcionamiento. Este proceso debió efectuarse de forma cíclica hasta alcanzar el estado óptimo.

Capítulo 5: Descripción detallada de la solución

En este capítulo se expone cómo se llegó a determinar el funcionamiento del servicio de mensajería en telefonía fija y como se realizó la selección final. Además se explica como se realizó el desarrollo de la solución.

5.1 Análisis de soluciones y selección final

Una de las primeras etapas que se dieron en el desarrollo del proyecto fue la búsqueda de cómo se conforma el servicio de mensajería en telefonía fija. Primeramente se ubicó países y compañías que brindaran este servicio. Aquí se observó que existen dos estándares internacionales, estos son:

- El estándar europeo, definido por la ETSI el cual es usado en compañías como British Telecom.
- El estándar americano ADSI, utilizado en Estados Unidos.

Ambos estándares, proponen un sistema de tres bloques para el servicio de mensajería en telefonía fija, los elementos que lo conforman son: la terminal del usuario, la red telefónica y el centro de mensajería. En la siguiente figura se muestra el sistema:



Figura 5.1.1 Diagrama de bloques del sistema de mensajería en telefonía fija

Entre estos dos estándares se seleccionó el europeo, ya que los equipos actuales del ICE trabajan con estándares europeos.

El estándar ETSI, posee dos protocolos para el servicio de mensajería en telefonía fija, estos son el Protocolo 1 y Protocolo 2. La diferencia entre estos protocolos es que uno está orientado exclusivamente hacia telefonía fija (Protocolo 2), mientras que el Protocolo 1 posee mayor compatibilidad con el protocolo usado para la mensajería de texto en telefonía móvil (GSM). Debido a la importancia de poder integrar, en un futuro, la mensajería de texto de telefonía fija con la de celular, se seleccionó el protocolo 1.¹

Ya que las terminales telefónicas existen en el mercado, el proyecto se enfocó en el desarrollo del centro de mensajería (SM-SC). El cual soporta el protocolo 1 de la ETSI.

Finalmente, se realizó una investigación acerca de los equipos con los que el ICE contaba para poder implementar el SM-SC; después de dicha investigación, se eligió una plataforma de valor agregado de la empresa APEX. La principal característica de esta plataforma que determinó su selección es que a través de ésta se podían alcanzar todos los requisitos físicos que determinaba el protocolo 1; por ejemplo: está conectada a red telefónica por medio de un enlace primario, por lo tanto cumple con la función de puerta de entrada del SM-SC y además es posible realizar la modulación y la demodulación de la información en FSK.

5.2 Descripción del hardware

Para implementar el centro de mensajes, se contó con una plataforma para servicios telefónicos (también llamadas “Plataformas de valor agregado”). Esta plataforma está compuesta básicamente por un servidor (computador) que se conecta a la red telefónica a través de tarjetas *Dialogic* diseñadas para aplicaciones telefónicas, que poseen dos enlaces primarios con protocolos de señalización ISDN, llamados E1 con velocidad de 2 Mbps. Cada uno de estos enlaces permite atender hasta 30 llamadas simultáneas, lo cual permitiría el envío o recepción de hasta 30 mensajes de texto de forma simultánea.

¹ Las características de este protocolo se exponen en el marco teórico.

Esta plataforma cuenta con 2 tarjetas Dialogic: una de ellas (D/300JCT) realiza la función del *gateway*¹ del SM-SC y es la que posee los dos enlaces de tipo E1. La otra tarjeta (DM/V 3600BP) es la que maneja los recursos de voz, los cuales permiten realizar la modulación y demodulación de señales en FSK, empleando el canal de voz. Ambas tarjetas son de tipo PCI y están montadas sobre el mismo servidor. Una parte importante del proyecto fue la configuración de las dos tarjetas, para lograr esto se investigó con los fabricantes (Intel) y se obtuvo un conjunto de librerías para el desarrollo, estas librerías son las *Global Call API (Application Programming Interface)* y a través de estas es posible configurar las tarjetas utilizando Visual C++.

5.3 Descripción del software

El software desarrollado en el centro de mensajería es estructurado en dos bloques, el encargado de la recepción de mensajes de texto y el que realiza la entrega de éstos. En ambos casos la aplicación del servicio se implementó utilizando el software OmniView de APEX, y para el desarrollo de las rutinas del protocolo 1 se utilizó Visual C++.

5.3.1 Descripción del software para la recepción de mensajes

Esta es la primera etapa que se da cuando un mensaje de texto es sometido, donde la terminal origen realiza una llamada hacia el SM-SC. En el momento que ingresa la llamada a la plataforma, OmniVox (de APEX) levanta la aplicación correspondiente a este servicio.

Descripción de la aplicación en OmniView

Utilizando OmniView es posible crear el flujo de la aplicación, éste es un software de diseño GUI (interfase de usuario gráfica), la forma de programación consiste en arrastrar iconos y realizar los enlaces entre éstos. Posteriormente se configura cada icono al agregar o quitar características.

¹ Ver Figura 3.1.1 del marco teórico

El flujo para la recepción de mensajes se muestra en la siguiente figura.

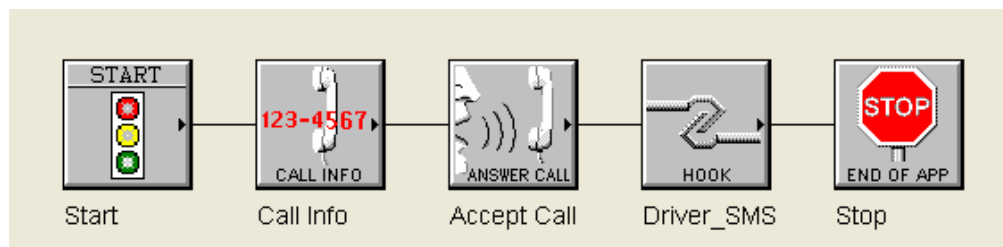


Figura 5.3.1.1 Flujo de la aplicación para la recepción de SM

Cuando OmniVox levanta la aplicación se empieza a ejecutar el *Start*, en este momento OmniVox ha asignado un canal de la tarjeta de red a la aplicación y ha obtenido información sobre la llamada entrante, como lo es el número de teléfono origen (ANI). Para poder acceder a esta información se debe utilizar otro icono llamado *Call Info*, en la siguiente figura se muestra como es enlazado el icono *Start* con el *Call Info*.

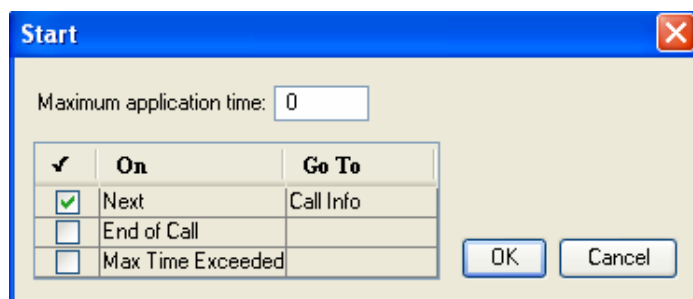


Figura 5.3.1.2 Configuración del icono *Start*

Para obtener la información del icono *Call Info* se debe crear variables donde se almacenan los datos requeridos. Estas variables corresponden a *ANI_Var* que almacena el número de teléfono origen, y *Port_Var* donde se guarda el número de puerto asignado a esa llamada. Estas dos variables son utilizadas más adelante dentro de la función *C-Hook*. En la siguiente figura se observa la forma de asignación de las variables:

Call Info

Name:

Operation

☒ Get
☐ Set

Port Selection

☒ Incoming port
☐ Specified port ID:

Parameter	Data
Digit Buffer5 (ANI)	@ANI_Var
Port ID	@Port_Var

✓	On	Go To
<input checked="" type="checkbox"/>	Success	Accept Call
<input type="checkbox"/>	Error	

OK Cancel

Figura 5.3.1.3 Configuración del icono *Call Info*

Este icono se enlaza con *Accept Call*, como se observa en la figura anterior. La función de *Accept Call* es de aceptar y responder la llamada, con este comando se establece el canal de voz de la llamada y la tarjeta de red. La configuración se muestra a continuación:

Answer Call

Name:

Settings

☒ Accept call
☒ Answer call
☐ Reject call if ANI is not available

✓	On	Go To
<input checked="" type="checkbox"/>	Success	Driver_SMS
<input type="checkbox"/>	Error	

OK Cancel

Figura 5.3.1.4 Configuración de la función *Accept Call*

El siguiente icono en el flujo de la aplicación corresponde al *C-Hook*, llamado *Driver SMS*, a través de él es posible incorporar librerías de enlace dinámico (.dll) que se desarrollan en Visual C++. El protocolo de mensajería se desarrolla en su totalidad utilizando este recurso, la descripción detallada de la librería se realiza en la siguiente sección.

Para configurar este icono se debe colocar dos parámetros, el primero es el nombre de la librería (que es la desarrollada usando Visual C++), en este caso se llama *Driver_SMS*, y el segundo parámetro es la puerta de entrada de la librería, es decir la función que se desea acceder, llamada “Recepción Mensajes”.

También hay que seleccionar el modo de funcionamiento, ya sea sincrónico o asincrónico. Se debe seleccionar el modo sincrónico, esto es que el flujo de la aplicación no continúa hasta que la función del *C-Hook* retorne. A partir del valor de retorno se puede enviar el flujo de la aplicación a que realice diferentes tareas, pero para este caso sin importar el valor de retorno la aplicación es enviada a que termine. Esto se logra al enlazarlo con el icono *Stop*, que libera los recursos utilizados en la aplicación.

La configuración del icono del C-Hook es la siguiente:

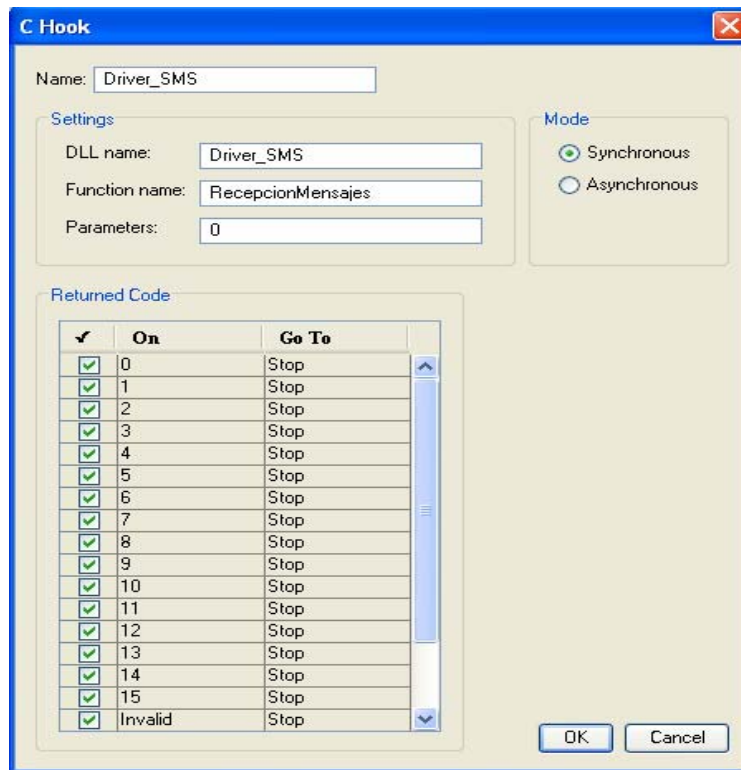


Figura 5.3.1.5 Configuración del icono *Driver SMS*

La librería que es llamada por este comando contiene las tres capas del protocolo 1 de la ETSI. También se encarga de guardar los mensajes de texto en la base de datos y genera un reporte de error en caso de que una transacción no se pueda efectuar.

Descripción de rutinas desarrolladas para el protocolo 1

En esta sección se expondrá las principales rutinas implementadas en la librería de enlace dinámica que se llama *Driver_SMS*. Estas rutinas son creadas utilizando el software Visual C++. Como el protocolo 1 de la ETSI se divide en tres capas, se desarrollo una clase independiente para cada una, así se logra trabajar en forma modular. Estas clases son:

- Clase Capa Física
- Clase Capa Enlace Datos
- Clase Capa Transferencia

Descripción de la Clase de Capa Física

Esta clase está más orientada al hardware, es la puerta de entrada (*gateway*) del centro de mensajería y además da servicio a la clase de Enlace de Datos. Esta clase es la encargada de controlar las tarjetas Dialogic. En la plataforma se encuentran dos de estas tarjetas, una que posee los recursos de red que permite conectarse al PSTN a través de un enlace primario (E1), y la otra que posee los recursos de voz que realiza la modulación y demodulación en FSK. Los tres procedimientos más importantes de esta clase son: “Inicializa Tarjeta”, “Recibe Trama” y “Transmite Trama”.

En el siguiente diagrama de flujo se muestra como debe ser utilizada esta clase:

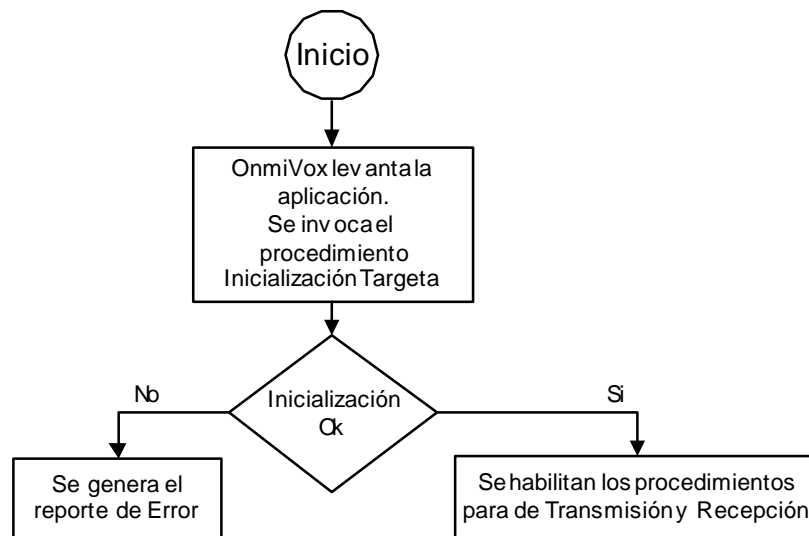


Figura 5.3.1.6 Diagrama de flujo de la clase “Manejo de Recurso de Voz”

Cuando se ejecuta el procedimiento “Inicializa Tarjeta” se obtienen dos manejadores, uno para la tarjeta de voz y el otro para la de red. Usando estos manejadores e instrucciones *Global Call API* se configuran los recursos analógicos de la tarjeta de voz para que pueda enviar y transmitir datos en FSK.

El procedimiento “Recibe Trama” lo que hace es escuchar el canal por el que entró la llamada para detectar si llegan datos en FSK. Si llegan datos son guardados en una dirección de memoria para que sean utilizados por la clase Capa de Enlace de Datos. Si después de 4 segundos no se han recibido datos el procedimiento genera un evento para terminar la conexión.

El procedimiento “Transmite Trama” lo que hace es leer una dirección de memoria y los datos almacenados en ésta son enviados a la red telefónica modulados en FSK.

Descripción de la Clase Capa Enlace de Datos

Las tres funciones principales de esta clase son:

- Dar servicio a la Clase Capa de Transferencia
- Validar las tramas recibidas de la Clase Capa Física
- Generar tramas que van a ser enviadas a la Clase Capa Física

El diagrama de flujo es el siguiente:

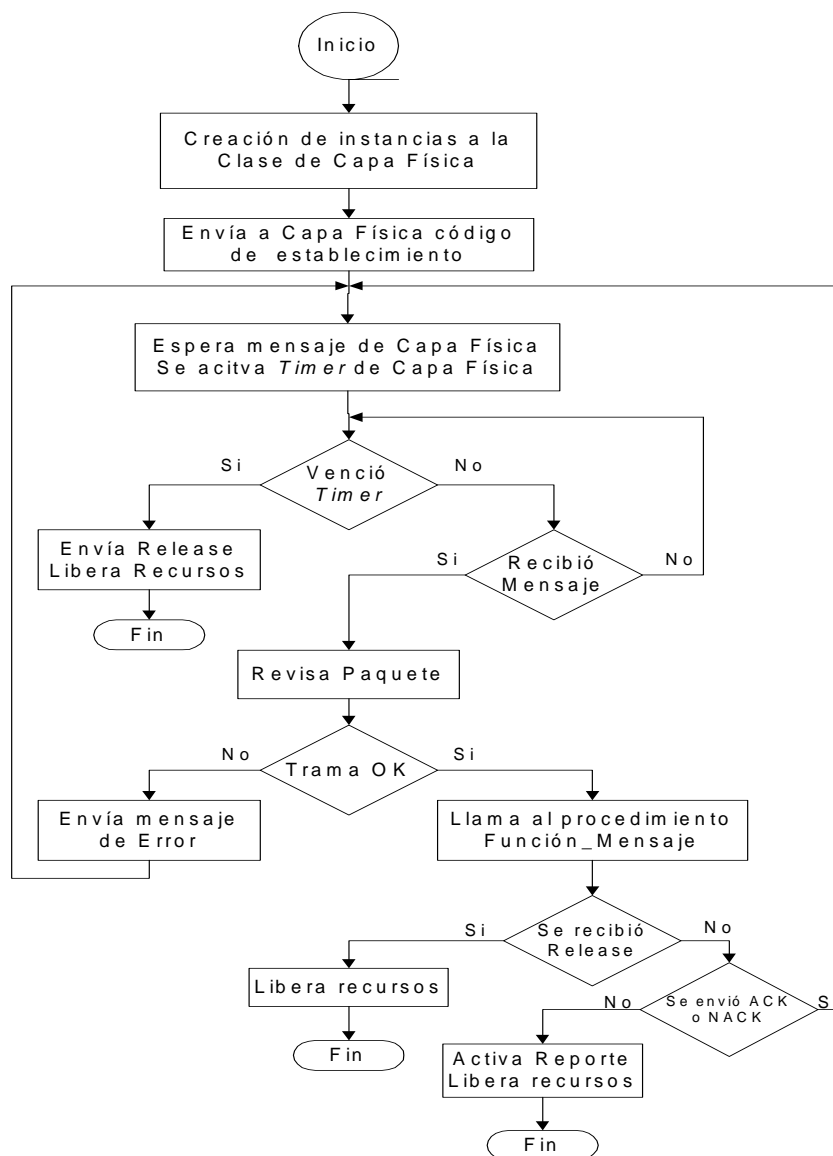


Figura 5.3.1.7 Diagrama de flujo del procedimiento “Capa Enlace Datos”

Lo primero que hace esta clase es enviar a la terminal de origen un código de establecimiento indicando que el SM-SC está listo para recibir el mensaje de texto. En este momento el flujo de la aplicación se queda esperando para recibir los datos de la terminal del usuario, si después de 4 segundos no se obtuvo respuesta se libera la conexión y se genera un reporte de error.

Si una trama es recibida dentro del tiempo permitido se invoca al procedimiento “Revisa Paquete”. Este procedimiento realiza las evaluaciones de la capa de enlace de datos para verificar si la trama recibida es correcta. Los tipos de error son clasificados en tres grupos, estos son:

- Tipo de mensaje desconocido
- Error en la longitud del mensaje
- Error de *checksum*

La evaluación es realizada en forma secuencial, y si se detecta un error se genera la trama con el código de error y es enviado hacia la termina origen. Para este caso, el tipo de mensaje siempre va a poseer el mismo valor (0x92H) indicado que el mensaje de la trama es un código de error y la longitud de mensaje que va a ser uno. La estructura de la trama sería de la siguiente forma:

DLL_SMS_ERROR = 0x92h	0x01h	Tipo error	<i>Checksum</i>
-----------------------	-------	------------	-----------------

Figura 5.3.1.8 Estructura para la trama error

Si la trama recibida no contiene errores de capa de enlace de datos, se llama al procedimiento “Función_mensaje”, este procedimiento lo que hace es clasificar la trama recibida. Si la trama recibida es de datos (que contiene en mensaje del usuario) se envía el paquete (PDU) a la clase de Capa de Transferencia. En este caso el SM-SC responde a la terminal con un código de reconocimiento o no reconocimiento para indicar si el mensaje de texto pudo o no ser guardado por el SM-SC.

Si la trama recibida es de error se vuelve a enviar la última trama que había envidado el SMCS, pero si el SM-SC recibe dos tramas de error consecutivas se genera un reporte de error y se liberarán los recursos.

Por último, si la trama recibida es de liberación indica que la transacción se ha completado y prepara al sistema para liberar los recursos.

Descripción de la Clase Capa De Transferencia

Esta clase da servicio a la aplicación y también toma decisiones a partir del PDU que recibe de la clase Capa de Enlace de Datos. El diagrama de flujo es el siguiente:

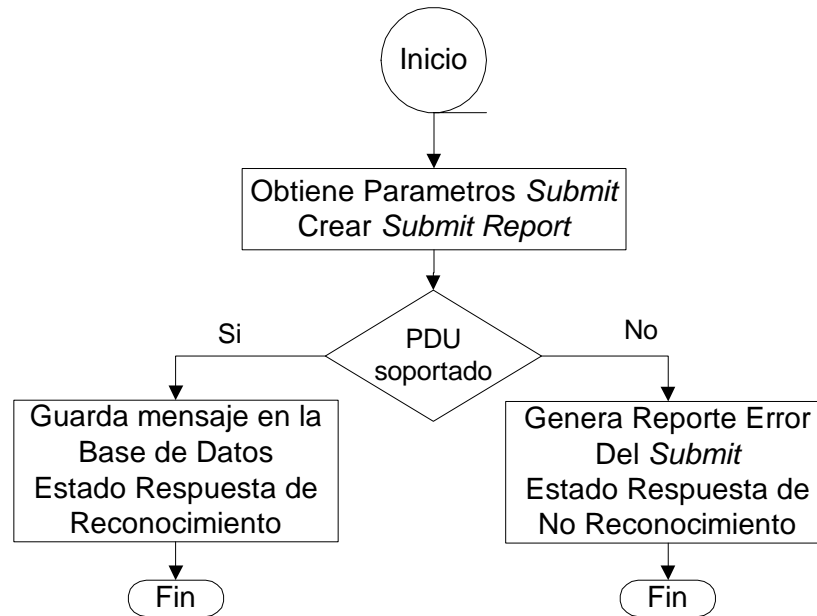


Figura 5.3.1.9 Diagrama de flujo de la clase Capa de Transferencia

Lo primero que realiza esta clase es invocar al procedimiento “Obtiene Parámetros *Submit*”, en éste se separa el paquete en los diferentes campos, por ejemplo: el mensaje del usuario, la dirección destino, esquema de codificación, etc.

Una vez que se han separado los elementos del paquete, se llama al procedimiento “Crear *Submit Report*”. En este procedimiento se analiza si el PDU recibido es soportado, por ejemplo: si el tipo de codificación de datos es válido. Aquí también se genera el paquete de respuesta para la terminal, para indicar si el mensaje de texto pudo ser procesado o no por el SM-SC. Si el PDU no es soportado, el paquete que se envía hacia la terminal origen llevará una cláusula fallo. Si el PDU es soportado, se guarda la información en una base de datos de Access para que pueda ser enviada a la terminal destino en el proceso del *Deliver*.

5.3.2 Descripción del software para la recepción de mensajes

La segunda etapa de servicio consiste en enviar el mensaje de texto, almacenado en la base de datos, a la terminal de destino. Para ello, el SM-SC debe hacer una llamada telefónica a la SM-TE de destino.

Descripción de la aplicación en OmniView

Este proceso se logra mediante aplicaciones virtuales; las cuales se pueden configurar para que cada cierto tiempo se activen y pedan ejecutar rutinas de OmniView. La secuencia que debe ejecutar se muestra en la figura 5.3.2.1.

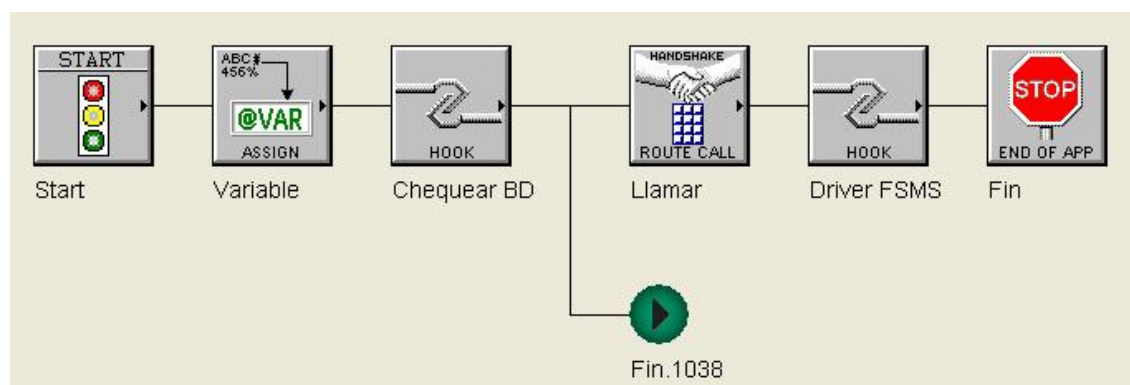


Figura 5.3.2.1 Flujo de la llamada para la entrega de mensajes de texto

Una vez que se activa la aplicación virtual, empieza a correr el flujo de la llamada (*Start*). Este primer ícono sólo indica el inicio del flujo de la llamada y se debe conectar al siguiente ícono (*Variable*), tal como se muestra en la figura 5.3.2.2.

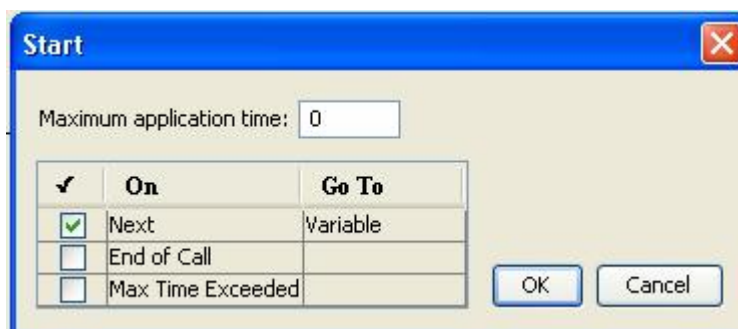


Figura 5.3.2.2 Configuración del *Start*

El ícono de *Variable* sirve para definir variables que serán usadas más adelante. En este caso, sólo se requiere definir la variable “@DNIS”, que servirá para guardar el número telefónico que se deberá marcar (número destino). En la figura 5.3.2.3 se observa la configuración de este ícono.

The screenshot shows a dialog box titled "Assign" with a close button (X) in the top right corner. Inside the dialog, there is a text field labeled "Name:" containing the text "Variable". Below this is a table with two columns: "Named Variable" and "Expression". The first row of the table contains "@DNIS" and "\"0\"". There are five empty rows below the first one. At the bottom of the dialog, there is a section with three columns: a checkmark column, an "On" column, and a "Go To" column. The "On" column has two rows: "Success" with a checked checkbox and "Error" with an unchecked checkbox. The "Go To" column has two rows: "Chequear BD" and an empty row. To the right of this section are "OK" and "Cancel" buttons.

Named Variable	Expression
@DNIS	"0"

✓	On	Go To
<input checked="" type="checkbox"/>	Success	Chequear BD
<input type="checkbox"/>	Error	

Figura 5.3.2.3 Definición de variables

En el siguiente ícono (*Chequear BD*) se ejecuta una rutina desarrollada en lenguaje C, que se encarga de revisar la base de datos donde se almacenan los mensajes recibidos por el SM-SC. La rutina que se ejecutará se llama “FuncionCheckBD” y se encuentra en el archivo “CheckBD.dll”, tal como se ve en la figura 5.3.2.4.

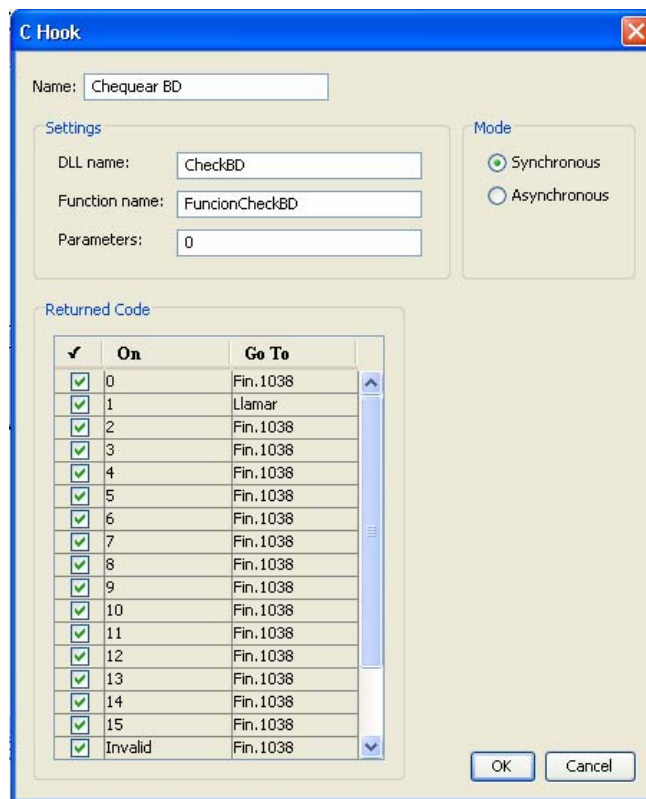


Figura 5.3.2.4 Configuración de *Chequear BD*

Si la rutina encuentra en la base de datos algún mensaje pendiente, tomará el número telefónico de destino de la misma base de datos y lo almacenará en la variable “@DNIS” (definida en el ícono anterior), luego retornará un “1” para continuar con el ícono *Llamar*. Si no se encuentra ningún mensaje pendiente para enviar, continuará con el ícono “Fin”, el cual finalizará la aplicación.

El ícono *Llamar* tiene la función de realizar la llama telefónica al número destino, este número fue previamente cargando en la variable “@DNIS”. Además se cargará en la variable “@PortID” el número del puerto por el cual se está haciendo la llamada. Esta configuración se ilustra en la figura 5.3.2.5.

Route Call

Name:

Route Mode

☒ Normal
☐ Initiate new call

Connect Options

☒ Before answer
☐ After answer
☐ No voice path routing

Destination

Type: ID:

Using	Data
Digit Buffer1 (DNIS1)	@DNIS
Digit Buffer2 (DNIS2)	
Digit Buffer3 (DNIS3)	
Digit Buffer4 (DNIS4)	
Digit Buffer5 (ANI)	234301511

☒ Wait for answer Cause code:
☐ Cancel digits (##) Port assigned:
☐ Call progress analysis Timeout (seconds):

✓	On	Go To
<input checked="" type="checkbox"/>	Success (Answer)	Driver FSMS
<input type="checkbox"/>	Busy	
<input type="checkbox"/>	Ring no answer	
<input type="checkbox"/>	Error	
<input type="checkbox"/>	Cancel Digits (##)	
<input type="checkbox"/>	Resource Group unavailable	

Figura 5.3.2.5 Configuración de la llamada

Después de realizar la llamada se establece el canal de voz y en este punto entran a funcionar las rutinas desarrolladas propias del protocolo de comunicación para la mensajería de texto (Protocolo 1). Estas rutinas están guardas en el archivo "FSMS.dll" y son llamadas por la función "FuncionFSMS", como se observa en la figura 5.3.2.6.

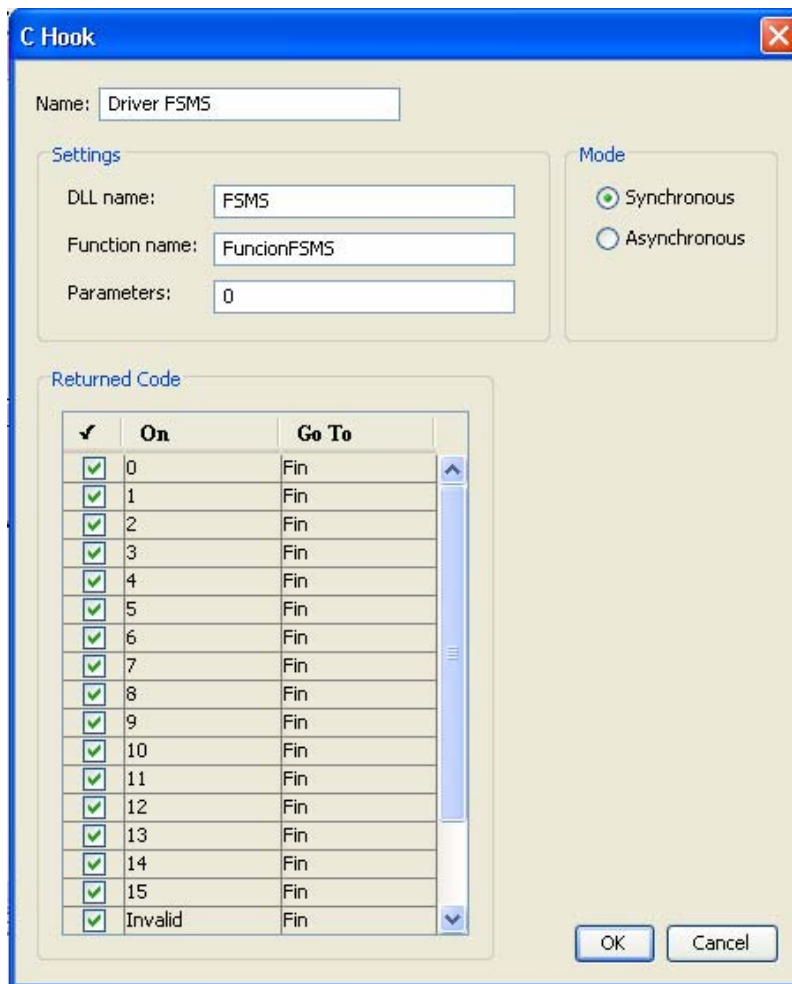


Figura 5.3.2.6 Configuración de *Driver FSMS*

Al finalizar las rutinas del protocolo se habrá podido entregar el mensaje de texto a su destinatario (en caso de un proceso exitoso) o se habrá podido determinar la causa de fallo (en caso de ocurrir algún error en la entrega). El ícono *Fin* marca la conclusión de la aplicación.

Descripción de rutinas desarrolladas para el protocolo 1

Para los procesos de recepción de mensaje se debe empezar por tomar la información de los diferentes PDU que conforman el paquete de la capa de transferencia (capa 3); esta información es tomada de una base de datos (*Access*), donde el mensaje de texto fue guardado previamente. Luego se

procede a armar el paquete de capa 3, la cual se envía a la capa de enlace de datos (capa 2); en esta capa se agregan los elementos de la trama que será enviada por el canal de voz.

En la capa de transferencia, las rutinas del *SMS-Deliver* y el *SMS-Submit* siguen una secuencia lineal tal como se muestra en diagrama de la figura 5.3.2.7, aunque los PDU son diferentes en cada uno de los casos.

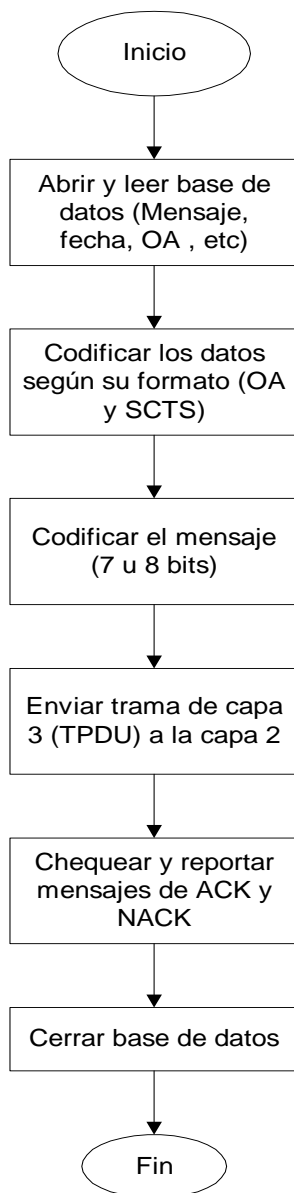


Figura 5.3.2.7 Diagrama de flujo de la capa de transferencia para las rutina de enviar mensaje

Lo primero que se debe hacer es leer la información de la base de datos (mensaje, dirección origen, hora, fecha, etc). Luego se configuran los datos según el formato que especifica el protocolo; además, se codifica el mensaje de texto, que por lo general se codifica en 7 bits.

Finalmente se arma el paquete de capa 3 con los PDU correspondientes, según sea el caso, y se envía a la capa de enlace de datos. El paquete del *SMS-Deliver* está compuesto por los siguientes elementos:

TP-MTI, TP-MMS, TP-RP, TP-UDHI, TP-SRI	TP-OA	TP-PID	TP-DCS	TP-SCTS	TP-UDL	TP-UD
--	-------	--------	--------	---------	--------	-------

Figura 5.3.2.8 Paquete del *SMS-Deliver*

El paquete del *SMS-Submit* es la siguiente:

TP-MTI, TP-RD, TP-VPF, TP-UDHI, TP-SRR	TP-MR	TP-DA	TP-PID	TP-DCS	TP-VP	TP-UDL	TP-UD
--	-------	-------	--------	--------	-------	--------	-------

Figura 5.3.2.9 Paquete del *SMS-Submit*

En el apéndice A.4 se explica cada uno de los parámetros que conforman los paquetes de la capa de transferencia.

Dentro de las rutinas más importantes que se han desarrollado en la capa de transferencia se encuentran: *Obtiene_TP_OA*, *Obtiene_TP_SCTS*, *Chequear_Deliver_Report*, *Codificador7bit*.

El procedimiento *Obtiene_TP_OA* permite obtener la dirección de origen con el formato que se especifica en el protocolo (ver formato de direcciones en el apéndice A.3). Esta dirección se toma de una base de datos y luego se le da el formato requerido.

El procedimiento *Obtiene_TP_SCTS* se emplea para configurar el tiempo de estampado (TP-SCTS) según el formato que especifica el protocolo. Al igual

que en la dirección de origen, hay que intercambiar los semi-octetos; en el siguiente cuadro se muestra un ejemplo para representar este parámetro.

Tabla 5.3.2.1 Representación del SCTS

	Binario	Hexadecimal
Año (ej: 2005)	0101 000	50
Mes (ej: agosto)	1000 0000	80
Día (ej: 10)	0000 0001	01
Hora (ej: 15)	0101 0001	51
Minutos (ej: 04)	0100 0000	40
Segundos (ej: 23)	0011 0010	32
Zona de tiempo	0100 1010	4A

Como se puede observar en el ejemplo, la zona de tiempo corresponde a A4h (sin intercambiar los octetos); esto se debe a que Costa Rica se encuentra en una zona donde el horario es -6 horas con respecto a GMT (*Greenwich Mean Time*). Las 6 horas corresponden a 24 cuartos de hora (0010 0100b o 24h) y se le debe agrega el bit de signo (1010 0100b = A4h), luego se realiza el intercambio de semi-octetos, quedando finalmente como 4A.

La rutina *Chequear_Deliver_Report* es una rutina de capa 3 que es llamada desde la capa 2 y se ejecuta cada vez que el SM-SC recibe un mensaje de *DLL_SMS_ACK* o *DLL_SMS_NACK*; permite almacenar los parámetros de estos mensajes en otra base de datos con el fin de verificar si el mensaje de texto enviado fue recibido por la terminal (en caso del *DLL_SMS_ACK*) o verificar el tipo de error que se dio (en caso de *DLL_SMS_NACK*).

La rutina *Codificador7bit* permite codificar el mensaje de texto (UD) en 7 bits. La codificación de 7 bits almacena cada uno de los caracteres del mensaje de texto en 7 bits en lugar de 8 bits, como se hace normalmente. Esto permite el ahorro de un byte cada 8 caracteres almacenados. Esta codificación se puede llevar a cabo ya que cada uno de los caracteres que conforman el alfabeto

corresponde a un código representado con un byte y se encuentran en un rango menor que 127 (código ASCII), por lo tanto, el octavo bit es siempre “0”. En la figura 5.3.2.10 se muestra el diagrama de flujo del decodificador de 7 bits.

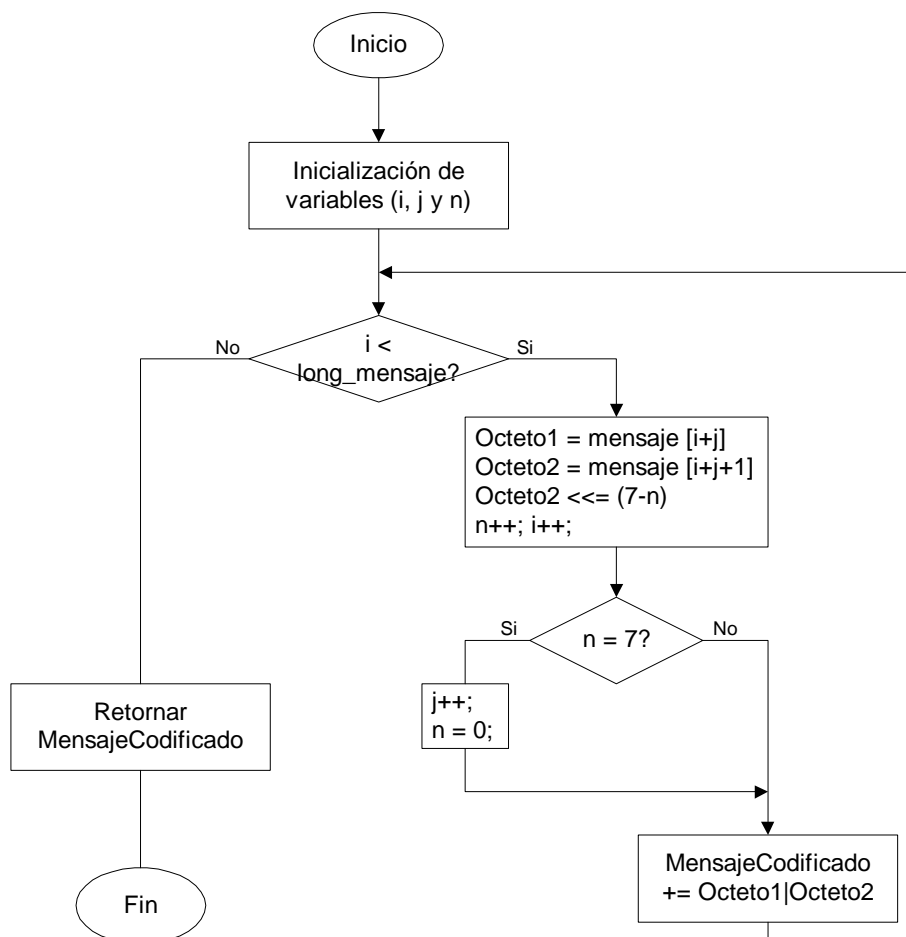


Figura 5.3.2.10 Diagrama de flujo del codificador de 7 bits

El codificador de 7 bits recibe como parámetro de entrada el mensaje de texto codificado en 8 bits (variable de tipo “string”) y ejecuta un ciclo que abarca toda la longitud del mensaje de texto. Finalmente el mensaje de texto codificado en 7 bits estará almacenado en la variable *MensajeCodificado*.

A nivel de capa de enlace de datos, el *SMS-Deliver* y el *SMS-Submit* y presentan el mismo tipo de mensaje: *DLL_SMS_DATA*. Esto se debe a que, en los 2 procesos, se lleva información (mensaje de texto) desde una entidad a otra. A su vez, esta característica hace que tengan el mismo diagrama de flujo para el traspaso de información. En la figura 5.3.2.11 se muestra el diagrama de flujo, de capa 2, para las rutinas de envío de mensaje.

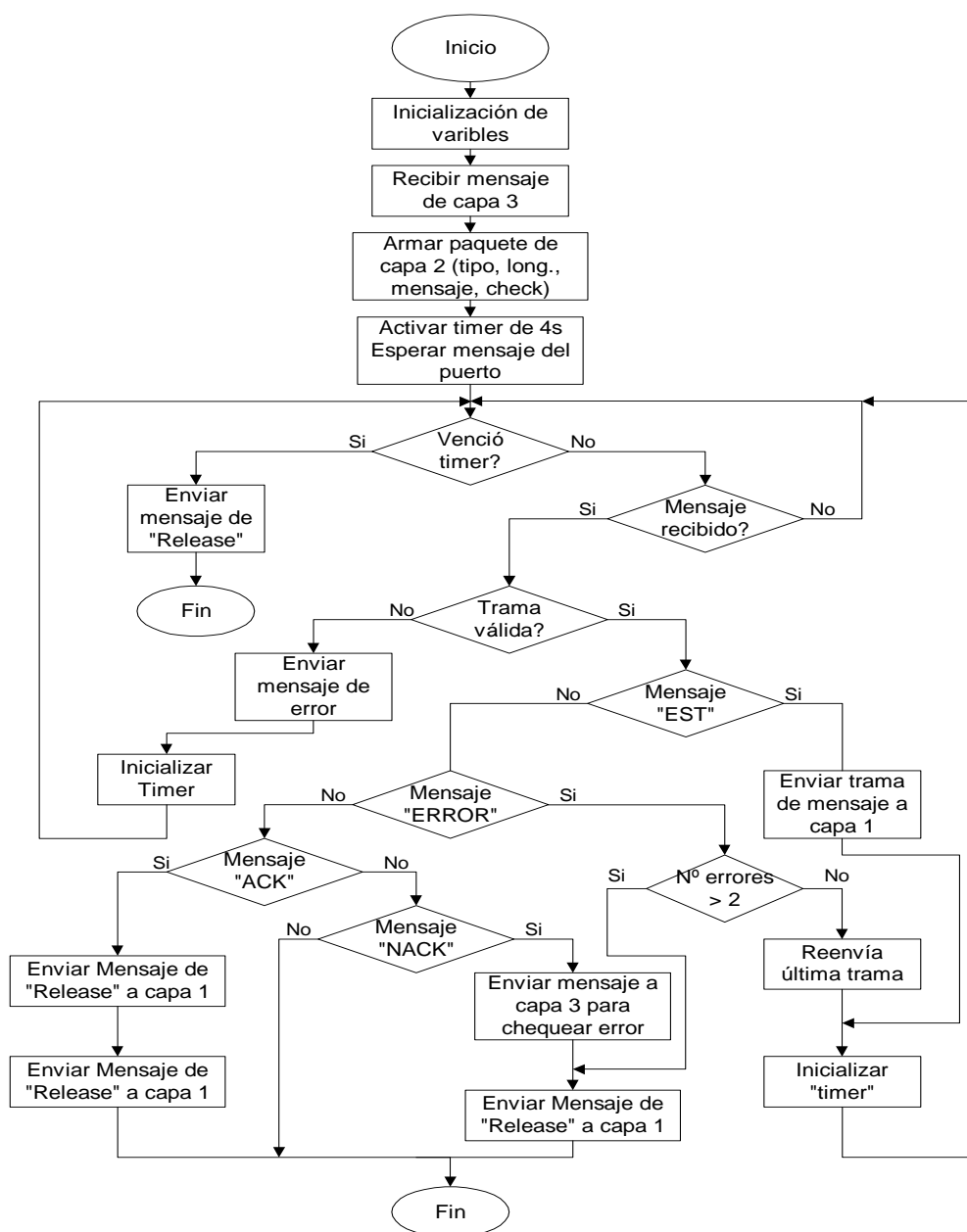


Figura 5.3.2.11 Diagrama de flujo de la capa de enlace de datos, para las rutinas de envío de mensaje

El proceso de capa 2, se realiza todo traspaso del mensaje de texto entre las entidades. En este proceso se contemplan las diferentes posibilidades que se pueden dar, tales como: mensajes de error, establecimiento, reconocimiento y no reconocimiento. Además, se maneja el temporizador para controlar el envío de tramas.

En la capa de enlace de datos se desarrollaron varias rutinas entre las cuales destacan: *Arma_Trama_Dato*, *Revisa_Paquete* y *Arma_Trama_Error*.

En la rutina *Arma_Trama_Dato* se ensambla la trama que contiene el mensaje de texto (*DLL_SMS_DATA*), este procedimiento recibe como parámetro de entrada el paquete de capa 3 (PDU). Si el paquete de capa 3 es mayor que 176 octetos, se segmentará en 2 o más paquetes para poder ser enviada.

La rutina *Revisa_Paquete*, se encarga de buscar errores en las tramas que recibe el SM-SC. Este procedimiento devuelve un valor entero, si no encuentra errores en la trama devuelve un "0", si el tipo de mensaje recibido es desconocido devolverá "3", para una longitud de mensaje incorrecta devuelve "2" y si detecta un *checksum* incorrecto devolverá "1". Los valores que retorna este procedimiento están basados en los tipos de error que se pueden dar en la capa 2.

El procedimiento *Arma_Trama_Error* se ejecuta si se ha detectado algún error en la trama que se recibió y toma como parámetro de entrada el valor devuelto por el procedimiento *Revisa_Paquete*. De este modo, se envía un mensaje de tipo *DLL_SMS_ERROR* con el valor correspondiente al error encontrado en la trama recibida.

Es importante mencionar que todos los procedimientos desarrollados están distribuidos en clases. Esto quiere decir que cada capa del protocolo posee una clase en la que se definen todos los procedimientos y variables propios de cada capa. Esto permite mayor facilidad a la hora de detectar errores y de depurar las rutinas desarrolladas.

Capítulo 6: Análisis de Resultados

En este capítulo se muestran los resultados de las mediciones y pruebas efectuadas para el envío y recepción de mensajes de texto entre el centro de mensajes y la terminal del cliente. Además se hace un análisis de estos resultados comparándolos con lo que se especifica en la teoría. Finalmente se realiza un análisis general del proyecto.

6.1 Medición de las frecuencias de transmisión y recepción

Entre las primeras mediciones que se realizaron están la de las frecuencias de transmisión y recepción utilizando un osciloscopio. Estas mediciones se realizaron para verificar que la configuración realizada en la tarjeta es correcta, y de este modo poder recibir y transmitir información en FSK. El protocolo 1 de la ETSI define que las frecuencias para la modulación en FSK son de $1300 \text{ Hz} \pm 1.5\%$ para un “1” lógico y de $2100 \text{ Hz} \pm 1.5\%$ para el “0” lógico.

El siguiente oscilograma muestra la medición de un “1” lógico

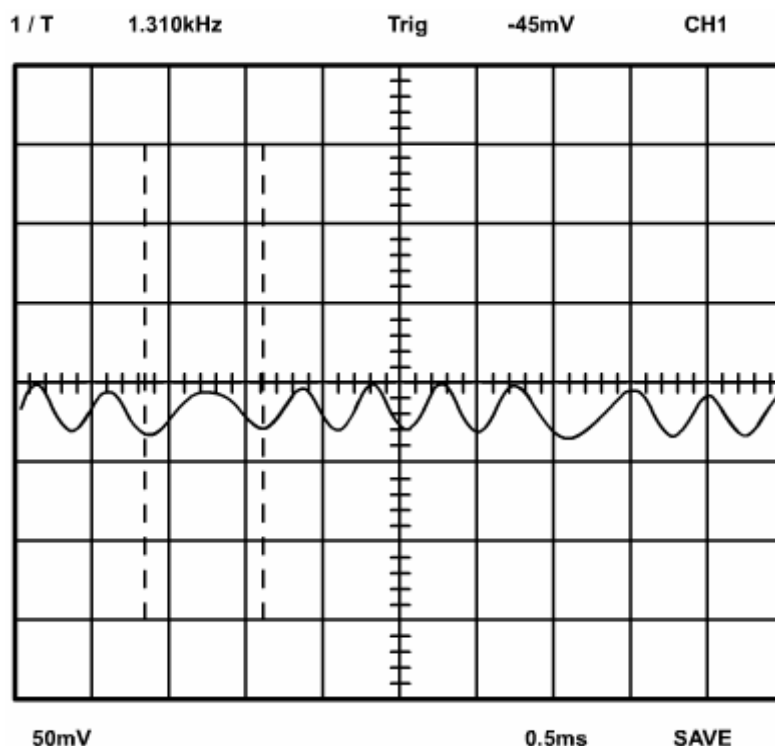


Figura 6.1.1 Medición de frecuencia para los bits de marca (“1” lógico)

En el oscilograma anterior se tiene que la frecuencia del “1” lógico es de 1310 Hz, y la frecuencias permitidas para el uno lógico tienen que estar entre el rango de 1319.5 Hz y 1280.5, la configuración de la frecuencia para los bits de marca esta correcta. En el siguiente oscilograma se muestra la medición de un “0” lógico:

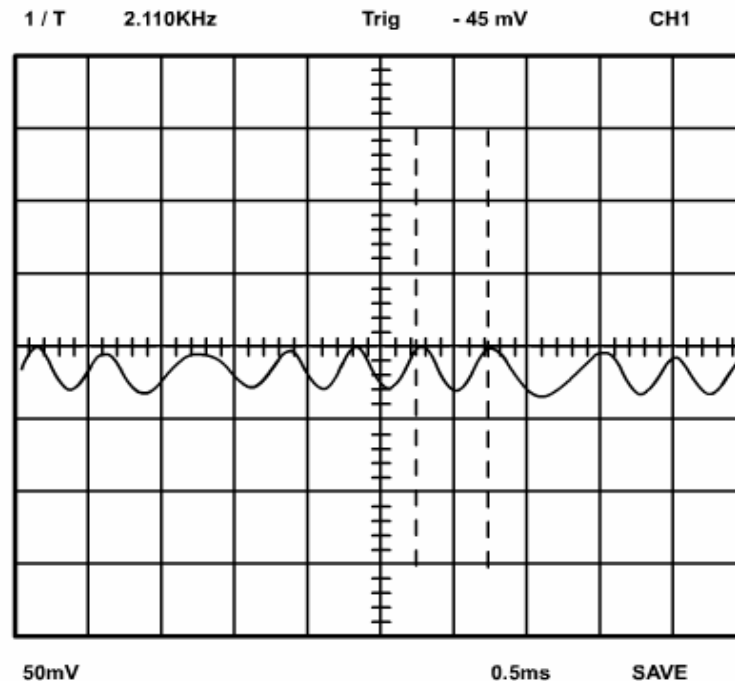


Figura 6.1.2 Medición de frecuencia para los bits de espacio (“0” lógico)

Del oscilograma anterior se tiene que la frecuencia para el cero lógico es de 2110 Hz, y los valores permitidos por el protocolo están entre el rango de 2131.5 Hz y 2068.5 Hz. Por lo tanto, la configuración de la frecuencia para los bits de espacio es correcta.

La configuración de las tarjetas se realizan dentro del *C-Hook* (rutinas desarrolladas en C++) utilizando instrucciones *Global Call API* de las tarjetas *Dialogic*. Como se puede observar en las figuras antes mencionadas, las frecuencias están dentro del rango establecidos para el uno y cero lógico, dando validez a que las configuraciones de las tarjetas tanto la de red como la de recursos de voz están correctas.

6.2 Proceso SMS_SUBMIT

Los resultados que se muestran en esta sección pertenecen a la transmisión de un mensaje corto de texto (*SMS_SUBMIT*) desde la terminal (SM-TE) hacia el centro de mensajería (SM-SC) en el caso normal. El siguiente flujo corresponde al intercambio de tramas al nivel de capa de enlace de datos de la terminal y el centro de mensajería.

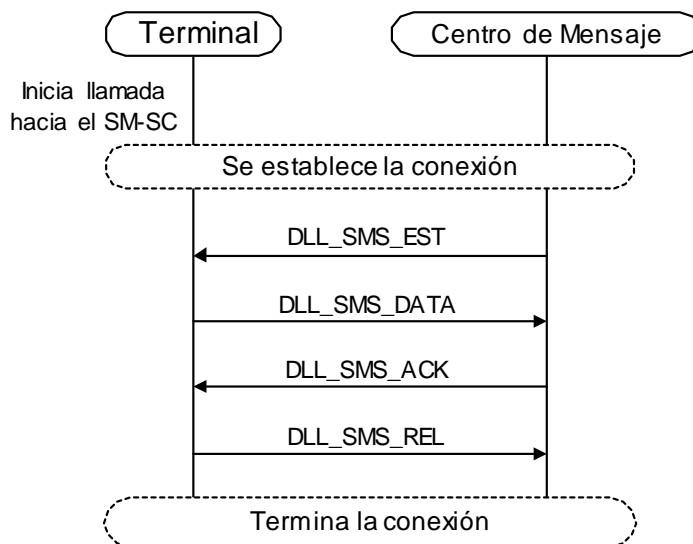


Figura 6.2.1 Diagrama de flujo de un SMS-Submit en caso normal

Para obtener el valor de las tramas de cada transacción se desarrolló una rutina en el centro de mensajes que estaba escuchando siempre el canal de voz. Tanto las tramas recibidas como enviadas por el centro de mensajes fueron guardadas en un archivo que almacena el código hexadecimal de dichas tramas. En la siguiente tabla se muestran los resultados obtenidos:

Tabla 6.2.1 Tramas recibidas y enviadas de un *SMS_SUBMIT*

Nº trama	Origen	Resultado de la trama (Hexadecimal)
1	SM-SC	93 00 6D
2	SM-TE	91 12 21 02 07 81 52 86 54 F2 00 F1 07 45 62 35 28 25 3E 01 34
3	SM-SC	95 09 01 00 50 21 10 61 31 33 4A D1
4	SM-TE	94 00 6C

A continuación se analizará cada una de las tramas. Después de que se estableció el canal de voz, el centro de mensajería envía la primera trama a la terminal informando el establecimiento de la conexión. En la siguiente figura se divide la trama según los elementos de la capa de enlace de datos:

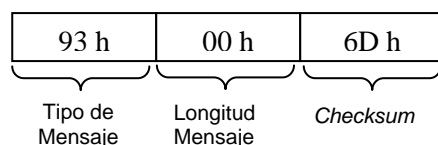


Figura 6.2.2 Trama *DLL_SMS_EST* enviada por el SM-SC

Esta trama es considerada 'nula' porque no presenta el campo del "mensaje"; por lo tanto, la trama del establecimiento se conforma de tres bytes. El tipo de mensaje indica que es de establecimiento, la longitud de mensaje es 00 h, debido a que no existe el campo de "mensaje", y finalmente el *checksum* que servirá para la validación de la trama.

Cuando la terminal recibió el establecimiento, responde con la trama que contiene el mensaje del usuario (trama número 2). En la siguiente figura se ilustra la representación según la capa de enlace de datos:

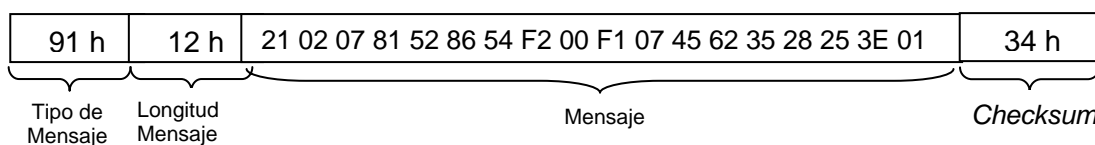


Figura 6.2.3 Trama *DLL_SMS_DATA* enviada por el SM-TE

El tipo de mensaje (91 h) indica que es una trama que contiene el dato del usuario (mensaje de texto). La longitud del mensaje indica el número de bytes presentes en el campo del mensaje. En el campo del mensaje está el PDU que se envían a la capa de transferencia, aquí se encuentran elementos de configuración, de información y el mensaje de texto. La siguiente tabla muestra la decodificación del paquete:

Tabla 6.2.2 Decodificación del PDU recibido en el *DLL_SMS_DATA*

Octeto (hexa)	Descripción
21	Primer octeto del <i>SMS-SUBMIT</i> (configuración del MTI)
02	Marca del mensaje (TP-MR)
07 81 52 86 54 F2	Dirección destino (TP-DA)
00	Identificador del protocolo (TP-PID)
F1	Esquema de codificación del dato (TP-DCS)
07	Longitud del mensaje del usuario (TP-UDL)
45 62 35 28 25 3E 01	Mensaje del usuario (TP-UD)

El primer octeto de cualquier PDU es usado para configuración, además de indicar si hay campos opcionales en el resto del paquete. En este caso, el valor recibido es un 21h indicando lo siguiente; el PDU es de un *SMS_SUBMIT*, el mensaje del usuario (TP-UD) no lleva encabezado y el campo opcional del periodo de validación (TP-VPF) no esta presente.

El siguiente byte es la marca del mensaje (TP-MR), este valor es usado junto al número origen (ANI) por el centro de mensajería para rechazar mensajes duplicados.

El campo de la dirección destino (TP-DA) está conformado por 6 bytes. El primero indica la longitud de número destino, en este caso es de 7 dígitos. El segundo byte es el tipo de dirección (81h) que indica el plan de numeración telefónico normal. Los bytes restantes son el número telefónico en semi-octetos (52h 86h 54h F2h), el número destino es: 256 84 52.

El esquema de codificación (TP-DCS) es F1h, lo que indica que el mensaje está codificado en 7 bits.

La longitud de mensaje (TP-UDL) es 07h, por lo tanto en el espacio del mensaje del usuario (TP-UD) debe haber 7 caracteres. El mensaje de texto codificado es: "EDUARDO".

El centro de mensajería evalúa la trama recibida y el paquete y al estar correctos responde con un reconocimiento positivo (trama N° 3 de la tabla 6.2.4). La representación de esta trama en la capa de enlace de datos es la siguiente:

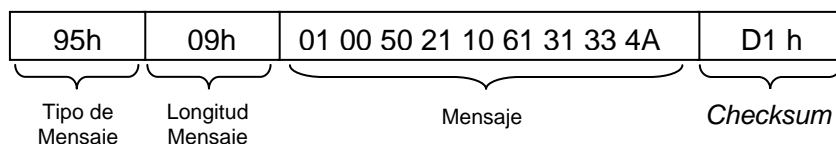


Figura 6.2.4 Trama *DLL_SMS_ACK* enviada por el SM-SC

El tipo de mensaje es 95h que indica que la trama corresponde a una confirmación positiva. El campo del mensaje es decodificado en la siguiente tabla:

Tabla 6.2.3 Decodificación del PDU recibido en el *DLL_SMS_ACK*

Octeto (hexa)	Descripción
01	Primer octeto del <i>SMS-SUBMIT</i> (configuración del MTI)
00	Identificador de parámetros (TP-PI)
50 21 10 61 31 33 4A	Tiempo de estampado (TP-SCTS)

El primer byte es de configuración, el valor de 01h indica que el MTI es de la respuesta de un *Submit*. El siguiente byte es el identificador de parámetros (TP-PI), el valor de 00h indica que ningún campo opcional está presente en el paquete.

Los últimos bytes corresponden al tiempo de estampado (TP-SCTS), esto es la hora en que el SM-SC recibió el mensaje de texto. En este caso, el mensaje fue recibido el 21/10/05 a las 13:33.

La última trama es enviada por la terminal, ésta corresponde a la liberación (trama N° 4 de la tabla 6.2.5), en la siguiente figura se muestra la estructura:

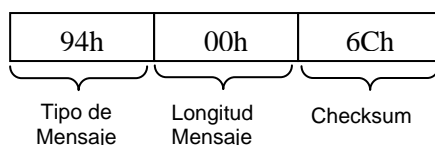


Figura 6.2.5 Trama *DLL_SMS_REL* enviada por el SM-TE

Esta trama es muy similar a la trama de establecimiento, de igual modo es considerada una trama 'nula' y lo importante es el tipo de mensaje (94h). Después

de enviar esta trama ambas entidades (el SM-SC y el SM-TE) liberan la llamada finalizando la recepción del mensaje por parte del SM-SC.

6.3 Proceso *SMS_Deliver*

Al igual que para el proceso de *Submit* se realizaron varias mediciones de las tramas recibidas y las tramas transmitidas por medio de la plataforma. Éstas corresponden a tramas de la capa de enlace de datos, las cuales contienen tipo de mensaje, longitud del mensaje, mensaje (paquete de capa 3) y *checksum*. Las tramas que se envían y que se reciben en el proceso de *Deliver* se ilustran en la figura 6.3.1.

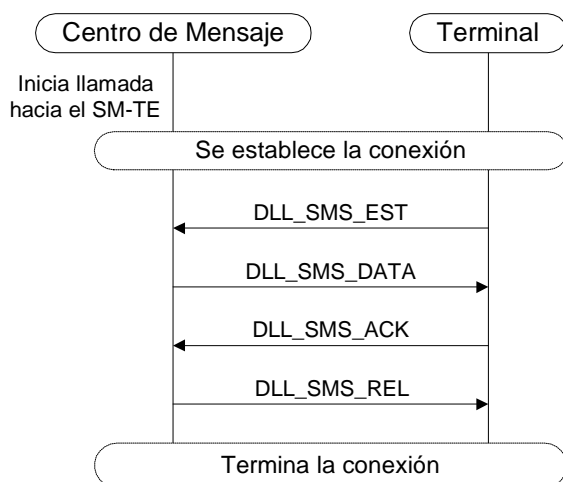


Figura 6.3.1 Tramas en el proceso de *Deliver* en caso normal

Las tramas fueron capturadas por medio de la plataforma y para efecto de pruebas se almacenaron en un archivo, en el cual se incluyó el código hexadecimal de cada uno de los bytes que componen la trama. En la tabla 6.3.1 se muestra un ejemplo de la entrega de un mensaje de texto, desde el centro de mensajes hacia la terminal del usuario.

Tabla 6.3.1 Resultados de las tramas del proceso de *Deliver*

Nº trama	Origen	Resultado de la trama (hexadecimal)
1	SM-TE	93 00 6D
2	SM-SC	91 16 04 82 23 68 F3 20 F1 50 21 70 01 34 53 42 07 45 62 35 28 25 3E 01 2E
3	SM-TE	91 02 00 00 6D
4	SM-SC	94 00 6C

Estos resultados son de un ejemplo específico; en este caso el mensaje de texto es “EDUARDO”.

El proceso de *SMS_Deliver* (entrega del mensaje de texto desde el SM-SC hacia el SM-TE) es similar al de *SMS_Submit*, solo que en este caso las tramas de establecimiento (*DLL_SMS_EST*) y reconocimiento (*DLL_SMS_ACK*) son enviadas por la terminal del usuario; mientras que las tramas del mensaje (*DLL_SMS_DATA*) y liberación (*DLL_SMS_REL*) son enviadas por el centro de mensajería (ver figura 6.3.1).

Las tramas de establecimiento, reconocimiento y liberación llevan la misma configuración que en el caso del *SMS_Submit*. La trama del mensaje corresponde a la trama Nº 2 de la tabla 6.3.1, dicha trama (de capa 2) está compuesta por los siguientes elementos:

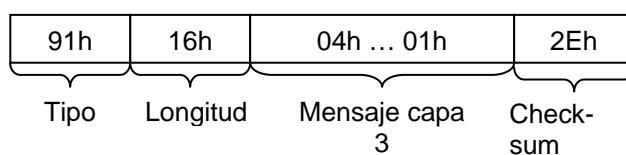


Figura 6.3.2 Trama del *Deliver*

El tipo (91h), la longitud (16h) y el *checksum* (2Eh) son elementos propios de la capa 2 que se emplean para comprobar la validez de la información recibida (mensaje de capa 3). Para el ejemplo analizado, esta información se ilustra en la figura 6.3.3, donde los valores representados están en hexadecimal.

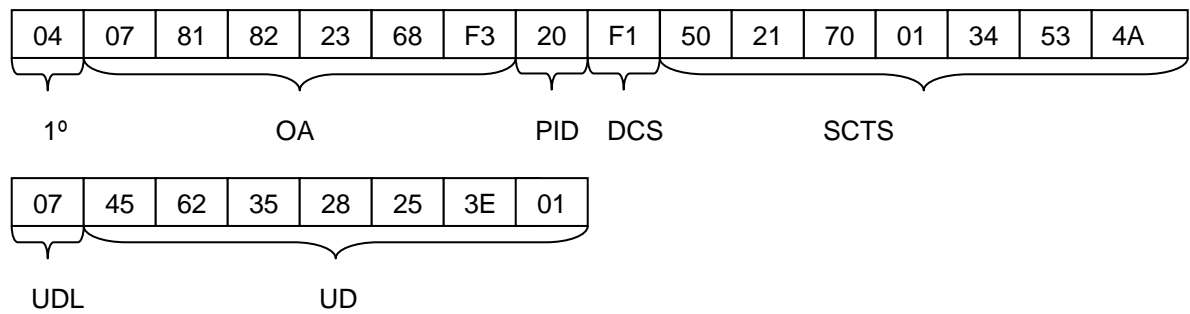


Figura 6.3.3 PDU del *Deliver*

El primer octeto (04h) indica que el paquete corresponde a un *Deliver*, no hay más mensajes por enviar y no lleva encabezado en el mensaje de texto. El primer octeto (07h) de la dirección de origen (OA) indica que el número origen es de 7 dígitos.

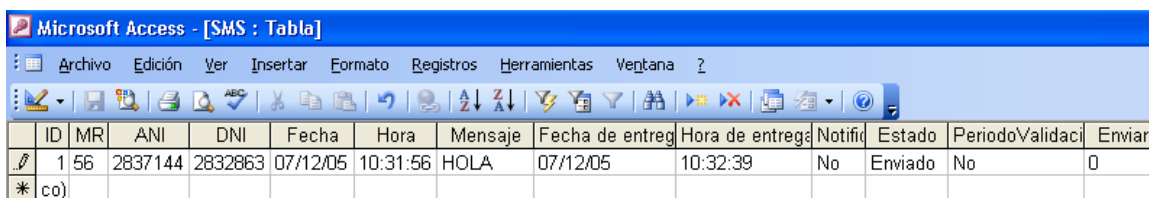
El segundo octeto de la dirección de origen indica el tipo de número telefónico y el plan de numeración; en este caso, el tipo de número es “desconocido”, este tipo de número es usado cuando el usuario o la red no tienen información acerca del plan de numeración y por lo tanto, se usa el plan de numeración telefónico normal (puede tener algún prefijo). Los siguientes octetos son el número telefónico (este número es el 283-2863); como se observa cada dígito del número telefónico corresponde a un semi-octeto los cuales se encuentran invertidos según el formato que especifica el protocolo¹.

El esquema de codificación (DCS) especifica que el mensaje de texto se encuentra codificado en 7 bits. Posteriormente vienen la longitud del mensaje de texto (UDL) y el mensaje de texto (UD), que como se dijo anteriormente, corresponde a “EDUARDO” codificado en 7 bits.

¹ Ver apéndice A.4

6.4 Base de datos

Para el desarrollo del proyecto se empleó una base de datos de Microsoft Access para almacenar temporalmente los mensajes de texto que llegan al centro de mensajes. En la figura 6.4.1 se observa el ejemplo para un mensaje de texto recibido y enviado por el centro de mensajes.



The screenshot shows the Microsoft Access application window titled 'Microsoft Access - [SMS : Tabla]'. The menu bar includes Archivo, Edición, Ver, Insertar, Formato, Registros, Herramientas, and Ventana. The toolbar contains various icons for file operations, editing, and viewing. Below the toolbar is a table with the following data:

ID	MR	ANI	DNI	Fecha	Hora	Mensaje	Fecha de entreg	Hora de entrega	Notifid	Estado	PeriodoValidaci	Enviar
1	56	2837144	2832863	07/12/05	10:31:56	HOLA	07/12/05	10:32:39	No	Enviado	No	0

Figura 6.4.1 Ejemplo de un mensaje guardado en la base de datos

La información que se encuentra almacenada en la base de datos de la figura anterior corresponde a un mensaje de texto que llegó al centro de mensajes desde un teléfono origen (ANI = 2837144) y fue entregado a su dirección de destino (DNI = 2832863).

Se escogió una base de datos de Access por motivos de simplicidad, pero como posible mejora del proyecto se podría utilizar una base de datos más robusta y especializada para este tipo de aplicaciones, por ejemplo MySQL. SQL es un lenguaje estandarizado y muy potente para consulta de bases de datos, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas. Otra de las razones para usar MySQL es la seguridad, ya que permite asignar diferentes permisos y privilegios a determinados usuarios para hacer las consultas o modificaciones de las tablas.

6.5 Análisis General del Proyecto

Para el desarrollo del proyecto “Desarrollo del Sistema de Mensajería para Telefonía Fija” se dieron varias etapas de investigación, se tuvo que estudiar cómo se brinda el servicio en otros países y estudiar los estándares que utilizan. Se encontraron dos estándares, el europeo y el americano, y debido a las

características del equipo del ICE se seleccionó el estándar europeo (ETSI). Una vez que se tenía claro el funcionamiento del servicio, se pudo determinar que el proyecto se enfocaría en el desarrollo del centro de mensajería, debido a que las terminales se podían conseguir en el mercado.

El estándar europeo tiene dos protocolos para este servicio, y con la visión de facilitar el enlace entre el servicio de mensajes entre los teléfonos móviles y fijos se seleccionó el protocolo 1.

Una vez que se definió qué protocolo utilizar, se obtuvo las características que debía tener el centro de mensajes, estas características ayudaron a determinar sobre cual equipo se implementaría el centro de mensajes. Después de estudiar varios equipos, se llegó a la conclusión que se desarrollaría en la plataforma de valor agregado de la compañía APEX, debido a que permite conectarse a la red telefónica por medio de un enlace primario usando señalización ISDN, además posee tarjetas Dialogic que permiten cumplir con los requisitos de la modulación en FSK y finalmente que es posible crear aplicaciones usando C++, esta última característica se utilizó para desarrollar el protocolo 1 de la ETSI.

Estas rutinas desarrolladas en C++ se estructuraron en clases, cada capa del protocolo se desarrolló en una clase aparte logrando con esto un trabajo modular. Cada clase está compuesta por los procedimientos necesarios que permiten configurar los elementos de cada capa. Por ejemplo, en la primera clase (capa física) están los procedimientos que configuran las tarjetas para que modulen en FSK.

El trabajo modular permitió detectar y corregir, de una forma más rápida, los errores en cada una de las capas. La depuración del código desarrollado se dio por medio de un proceso iterativo de pruebas, tanto de los procedimientos individuales de cada capa como de todo el conjunto.

En la capa física se logró controlar las tarjetas Dialogic de Intel utilizando las librerías *Global Call API*. Esto fue necesario ya que la herramienta de diseño de alto nivel (OmniView), no permite realizar la configuración que requería el protocolo. Por lo tanto, utilizando Visual C++ y las librerías *Global Call API* se pudo configurar los recursos analógicos para la modulación en FSK, enlazar las tarjetas (para compartir recursos) y controlar los canales de entrada y salida.

Una de las limitaciones que presenta el prototipo desarrollado está en el modo de entrega de los mensajes de texto. Existen dos modos de entrega, el primero consisten en que el centro de mensajes hace la llama a la terminal de destino y ésta, al detectar que la llama proviene del centro de mensajes (por medio de la información del identificador de llamadas "Caller ID"), contesta la llamada automáticamente y recibe el mensaje de texto. Para el otro modo entrega, el centro de mensajes llama a la terminal destino, pero ésta no contesta la llamada (de esta forma la terminal sabe que hay un mensaje de texto para ella), sino que posteriormente la propia terminal realiza la llamada hacia el centro de mensaje para recibir el mensaje de texto. La limitación del prototipo consiste en que sólo funciona el segundo modo de entrega, esto se debe a que la terminal de destino debe interpretar la información del "Caller ID"¹ (al número telefónico de la plataforma se le debe agregar 2 dígitos más: sub-dirección y modo de entrega). La modificación de esta información no se pudo llevar a cabo debido a aspectos de configuración en la conexión del E1 de la plataforma y de la central telefónica a la que está conectada, lo cual se sale del alcance del proyecto.

Al desarrollarse un centro de mensajería de texto en telefonía fija se deja abierta la posibilidad de conexión de ésta con los centros de mensajería en teléfonos móviles. Para ello es necesario crear un cliente con el centro de mensajería fija que maneje el protocolo SMPP; este protocolo es el que se usa para el intercambio de información entre centros de mensajes para telefonía móvil. De poder hacerse esta conexión sería posible enviar mensajes de texto desde teléfonos fijos hacia teléfonos móviles y viceversa.

¹ Ver figura A.3.2 del apéndice A.3

Capítulo 7: Conclusiones y recomendaciones

7.1 Conclusiones

- El uso de las instrucciones *Global Call API* permite el control de las tarjetas *Dialogic* en un bajo nivel de programación. Esto hace que las rutinas desarrolladas sean independientes de la herramienta de desarrollo (en este caso OnmiView) y puedan ser fácilmente usadas en otros programas.
- El uso de clases para el desarrollo de las rutinas del SMSC permitió separar cada capa del protocolo, y así obtener una solución modular y jerárquica.
- El uso de un estándar internacional para el desarrollo del proyecto hizo posible la utilización de teléfonos existentes en el mercado para comprobar correcto funcionamiento del servicio. Con esto garantiza la estandarización del servicio.
- Es posible desarrollar el servicio de mensajería de texto para telefonía fija con la infraestructura y equipo actual del ICE.

7.2 Recomendaciones

- Se podría usar una base de datos más robusta para el almacenamiento de los mensajes de texto, por ejemplo: MySQL. Una base de datos de este tipo permitiría un manejo más eficiente de los mensajes de texto, principalmente cuando el servicio funcione de forma masiva.
- Se podría conectar este centro de mensajería con los de móviles por medio de la creación de un cliente SMPP, para el intercambio de mensajes de texto entre teléfonos fijos y celulares. Esto le daría un atractivo mucho mayor al servicio.
- Se deben configurar adecuadamente las centrales telefónicas para poder operar con cualquiera de los 2 modos de entrega de mensajes de texto. Las centrales telefónicas manejan privilegios que permiten modificar ciertos parámetros tales como el número de identificador de abonado (ANI); al modificar este valor se podrá definir el modo de entrega del mensaje. En el primer modo de entrega el cliente destino no deberá pagar por recibir el mensaje; mientras que en el segundo modo de entrega, el cliente debe pagarlo ya que deberá hacer una llamada hacia el centro de mensajería.
- Es necesario realizar pruebas masivas para comprobar el buen desempeño del centro de mensajería. Mediante estas pruebas se podrá comprobar la eficiencia de las rutinas del protocolo y la aplicación desarrolladas.

Bibliografía

1. B. Vega Palacios. Sistemas de Señalización de Redes Telefónicas. ABCIET-ICI. Madrid 1992.
2. Commutation y transmission, ISDN. SOTELEC, Paris.
3. www.Telecom Asia Bringing SMS to the fixed line world - Advertorial - ZTE's ZXF118 integrated messaging platform.html Se obtiene información sobre las terminales de telefonía fija.
4. ETSI TS 123 040: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of Short Message Service (SMS) (3GPP TS 23.040 Release 5)".
5. ETSI EN 300 659-1: "Access and Terminals (AT); Analogue access to the Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 1: On-hook data transmission".
6. ETSI EN 300 659-2: "Access and Terminals (AT); Analogue access to the Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 2: Off-hook data transmission".
7. ETSI EN 300 659-3: "Access and Terminals (AT); Analogue access to the Public Switched Telephone Network (PSTN); Subscriber line protocol over the local loop for display (and related) services; Part 3: Data link message and parameter codings".
8. "Access and Terminals (AT); Short Message Service (SMS) for PSTN/ISDN; Short Message Communication between a fixed network Short Message Terminal Equipment and a Short Message Service Centre".

9. ETSI TS 100 900 V7.2.0: "Digital cellular telecommunications system (Phase 2+) (GSM); Alphabets and language-specific information (GSM 03.38 version 7.2.0 Release 1998) ".
10. ETSI TS 100 901 V7.4.0 : "Digital cellular telecommunications system (Phase 2+) (GSDM) ;Technical realization of the Short Message Service (SMS) (GSM 03.40 version 7.4.0 Release 1998)".

Apéndices

A.1 Abreviaturas

AES: Onmivox APEX Enhanced Switch

ANI: Identificador del número de abonado

DNI: Identificador del número de destino

DLL: Capa de enlace de datos (Data Link Layer)

DMI: Identificador del modo de entrega (Deliver Mode Identifier)

DTMF: Tonos de multifrecuencia

FSK: Conmutación de frecuencia (Frequency Shift Keying)

GMT: Hora de Greenwich (Greenwich Mean Time)

GSM: Sistema global para comunicación de celulares (Global System for Mobile communication)

GUI: interfase de usuario gráfica

ISDN: Red digital de servicios integrados (Integrated Services Digital Network)

ISO: Organización de Estandarización Intenacional (International Standard Organization)

PDU: Unidad de Descripción de Protocolo (Capa de transferencia)

PSTN: Red de telefonía pública (Public Switched Telephone Network)

SM: Mensaje corto de texto (Short Message(s))

SM-AL: Capa de aplicación de mensajes de texto (Short Message Application Layer)

SM-DLL: Capa de enlace de datos para mensajes de texto (Short Message Data Link Layer)

SME: Entidad para mensajes texto (Short Message Entity)

SMS : Servicios de mensajería corta de texto (Short Message Service)

SM-SC: Centro de mensajería de texto (Short Message Service Centre)

SM-TE: Equipo terminal para mensajes de texto (Short Message Terminal Equipment)

SM-TL: Capa de transferencia para mensajes de texto (Short Message Transfer Layer)

SS7: Sistema de Señalización N° 7 (Signalling System N° 7)

TL: Capa de transferencia (Transfer Layer)

A.2 Información sobre la institución

A.2.1 Descripción de la empresa

El Instituto Costarricense de Electricidad (ICE) fue el 8 de abril de 1949 como una institución autónoma, con personalidad jurídica y patrimonio propio. Está dotado de plena autonomía e independencia administrativa, técnica y financiera. Al ICE le corresponde, por medio de sus empresas, desarrollar, ejecutar, producir y comercializar todo tipo de servicios públicos de electricidad y telecomunicaciones, así como actividades o servicios complementarios a estos.

Como objetivos primarios el ICE debe desarrollar, de manera sostenible, las fuentes productoras de energía existentes en el país y prestar el servicio de electricidad. A su vez, se encarga de desarrollar y prestar los servicios de telecomunicaciones, con el fin de promover el mayor bienestar de los habitantes del país y fortalecer la economía nacional.

La misión del ICE es satisfacer las necesidades y expectativas evolutivas de los clientes y la sociedad costarricense, mediante el suministro oportuno de servicios y aplicaciones de telecomunicaciones e información de calidad, a precios y tarifas competitivos, con la tecnología adecuada y el mejor recurso humano.

Tiene como visión ser una empresa propiedad del Estado, competitiva de clase mundial, líder en el mercado de las telecomunicaciones e información, con la mejor tecnología y recurso humano al servicio del cliente y de la sociedad costarricense.

Actualmente el ICE en búsqueda de una mayor eficiencia en sus servicios se ha unido a un grupo corporativo de empresas estatales, integrado por el propio ICE (Sectores Electricidad y Telecomunicaciones) y sus empresas: Radiográfica Costarricense S.A. (RACSA) y la Compañía Nacional de Fuerza y Luz S.A. (CNFL), las cuales han trazado su trayectoria, mediante diversos proyectos de modernización desarrollados en las últimas décadas.

A.3 Procesos del Protocolo 1

El envío del mensaje (*SMS-Submit*, de la SM-TE hacia el SM-SC)

En esta sección se explicará el flujo normal en el envío de un mensaje de texto desde la terminal hacia el centro de mensaje. El primer paso es establecer la conexión con el SM-SC; para ello, la terminal del usuario descuelga la línea (automáticamente) y marca el número del SM-SC. Esto lo puede hacer por medio de pulsos o tonos (DTMF). En la siguiente figura se muestra la situación:

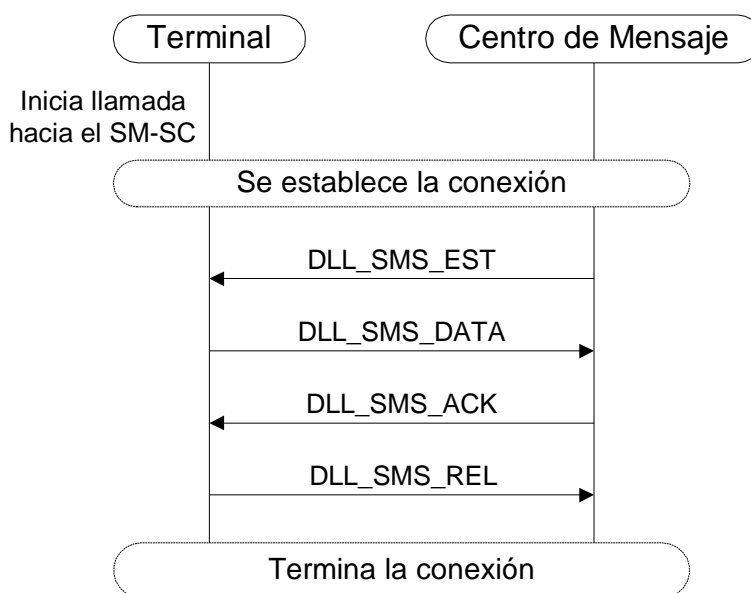


Figura A.3.1 Diagrama de flujo de un *SMS-Submit* en caso normal

Cuando la llamada es contestada por el SM-SC se establece el canal de voz y empieza la transmisión entre la terminal y el centro de mensajes. El SM-SC envía el primer paquete con el código de establecimiento (*DLL_SMS_EST*). Este paquete es para informar a la terminal del usuario que el centro de mensajes está listo para recibir los datos.

La terminal responde y envía el mensaje de texto, esta nueva trama lleva el código *DLL_SMS_DATA*. Una de las características de este protocolo es que cada trama que es enviada debe recibir una confirmación indicando el estado del paquete recibido, para el caso normal el centro de mensaje responde con el

código *DLL_SMS_ACK*. Por último, para liberar la línea, la terminal del usuario envía la trama de *DLL_SMS_REL*, dando fin a la transacción.

En caso de haber más de una SM-TE conectada a la misma línea telefónica de destino, el usuario que envía el mensaje puede direccionarlo agregando un número de subdirección (del 0 al 9) al final del número de teléfono destino (esto se hace cuando el usuario escribe el mensaje de texto).

Toda la información transmitida entre las entidades se hace por medio del canal de voz con modulación FSK.

Entrega del mensaje (*SMS-Deliver*, del SM-SC a la SM-TE)

Para entregar el mensaje a su destino, el SM-SC realiza una llamada a la SM-TE destino. La SM-TE interpreta el número del identificador de llamadas recibido (como se muestra en la figura 3.2.1.2) para identificar al SM-SC y decidir cómo debe recibir el mensaje de texto (existen 2 modos de entrega). Para habilitar esta interpretación, se deben guardar en la memoria de la SM-TE al menos un número de un SM-SC (pueden existir varios centros de mensajes).

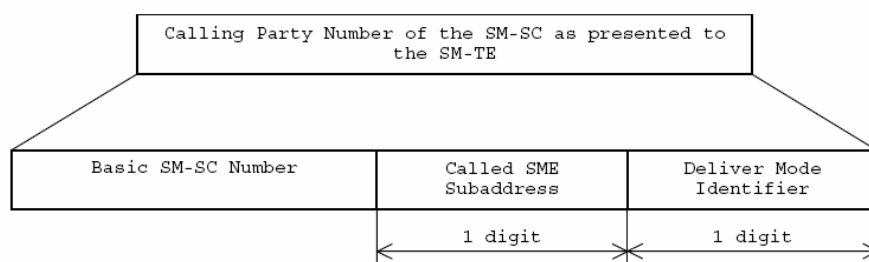


Figura A.3.2 Interpretación del identificador de llamadas en la SM-TE¹

Si la secuencia del número del SM-SC y la subdirección son iguales al valor guardado en la SM-TE, la llamada entrante corresponde a un mensaje de texto para esta terminal. En este caso, la terminal evalúa la forma en que se debe recibir el mensaje. El identificador del modo de entrega (ver figura 3.2.1.2) permite al

¹ Tomado de ETSI ES 201 912 V1.1.1 (2002-01)

SM-SC dos posibilidades de iniciar una conexión con la SM-TE. La diferencia entre estas dos posibilidades está en el cobro de la conexión.

La primera opción es que la SM-TE conteste la llamada del SM-SC después de que la información del identificador se haya transmitido y evaluado. En este caso, el costo de la conexión es cargado al SM-SC.

La segunda opción es que el SM-SC corte la conexión después de un corto tiempo. Este tiempo garantizará que la información del identificador de llamadas se haya transmitido a la SM-TE. Usando esta opción, no se responden las llamadas provenientes del SM-SC, sino que la SM-TE genera una llamada de regreso hacia el SM-SC para recibir el mensaje. En este caso, el costo de la conexión es cargado a la SM-TE.

A.4 Arquitectura del protocolo 1

La arquitectura del protocolo 1 de ETSI está compuesto por tres capas: La Capa Física (PL), la Capa de Enlace de Datos (DLL) y la Capa de Transferencia (TL).

Capa Física (PL)

Esta capa permite enviar y recibir mensajes de la capa de enlace de datos utilizando el ancho de banda del canal de voz. Toda comunicación que se dé entre la SM-TE y el SM-SC debe ser “half duplex”, por medio de modulación FSK dentro del ancho de banda del canal de voz (300 a 3400Hz) y con velocidad de 1200 Baud.

Las entidades (SM-SC y SM-TE) deberán ser capaces de recibir y enviar señales de FSK con las siguientes frecuencias:

- Marca = "1" lógico, frecuencia: 1 300 Hz \pm 1,5 %.
- Espacio = "0" lógico, frecuencia: 2 100 Hz \pm 1,5 %.

La entidad que contesta la llamada entrante debe enviar la primera trama de sincronización (*DLL_SMS_EST*) a la entidad que inició la llamada. En la figura A.4.1 se ilustra un ejemplo de la entrega de un mensaje de texto desde una SM-TE hacia un SM-SC.

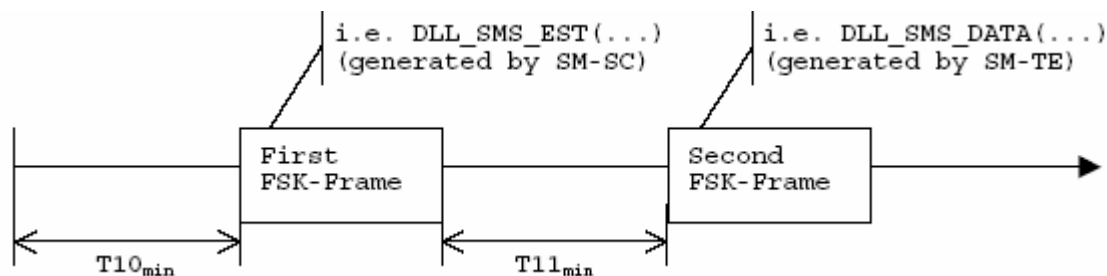


Figura A.4.1 Ejemplo de la entrega de un mensaje de texto¹

¹ Tomado de ETSI ES 201 912 V1.1.1 (2002-01)

Una vez que la conexión entre las dos entidades se establece, la primera trama (*DLL_SMS_EST*) en FSK es enviada por el SM-SC a la SM-TE. Posteriormente, la SM-TE responde con la segunda trama (*DLL_SMS_DATA*) que contiene el mensaje que se desea enviar.

Para asegurar que todos los elementos de la red están listos para transmitir la información de forma correcta, se debe dejar pasar un mínimo de tiempo (T_{10}) entre el momento en que se establece la conexión y el momento en el que se debe enviar la primera trama. Este tiempo deberá ser:

$$T_{10_{\min}} = n * 100ms ; \quad n = 1 \text{ a } 256$$

Para usar la comunicación “half duplex” en la red se debe activar el supresor o cancelador de ecos y debe haber un tiempo de espera mínimo entre dos tramas consecutivas de:

$$T_{11_{\min}} = 100ms$$

Capa de enlace de datos (DLL)

Esta capa es responsable de la detección de errores y del control del tiempo de respuesta de las entidades. Los mensajes de esta capa, al igual que los mensajes de la capa de transferencia, tienen como máximo un tamaño de 176 bytes. El formato del mensaje de la DLL se ilustra en la figura A.4.2.

Señal de marca	Tipo de mensaje	Longitud del mensaje	Mensaje (Mensaje de capa de transporte)	Check-sum
----------------	-----------------	----------------------	--	-----------

Figura A.4.2 Formato de la trama de la DLL

- Señal de marca:

Consiste de un bloque de 80 +/- 25 bits de marca (“1” lógico). Esto sirve para sincronizar el receptor de FSK de la entidad y se envían en ambos sentidos.

- Tipo de mensaje (1 octeto):

Este octeto contiene un valor codificado para identificar el mensaje. El bit más significativo es usado como bit de extensión; si es “0”, el mensaje es mayor que los 176 bytes que permite la trama y deberá esperar otra trama para completar el mensaje; si es “1”, el mensaje viene completo. Los códigos del Tipo de Mensaje se describen a continuación:

Tabla A.4.1. Tipos de mensajes en la capa de enlace de datos

Tipo de mensaje	Código binario	Descripción
DLL_SMS_DATA	X001 0001b	Lleva un SM dato
DLL_SMS_ERROR	X001 0010b	Indica que ha ocurrido un error en el DLL
DLL_SMS_EST	X001 0011b	La conexión del DLL ha sido establecida
DLL_SMS_REL	X001 0100b	Liberar la DLL conexión
DLL_SMS_ACK	X001 0101b	Lleva un reconocimiento positivo del SM
DLL_SMS_NACK	X001 0111b	Lleva un reconocimiento negativo del SM

- Longitud del mensaje (1 octeto):

Este octeto contiene un valor binario codificado que corresponde al número de octetos que tiene el mensaje de la DLL (excluyendo los octetos del tipo de mensaje, longitud del mensaje y el *checksum*). La longitud máxima del mensaje es de 176 octetos.

- *Checksum* (1 octeto):

Contiene el complemento a dos del módulo de 256 de la suma de todos los octetos de la trama, excluyendo el *checksum*.

En caso de que se produzca un error (*checksum* equivocado, un tipo de mensaje desconocido, etc), la DLL de la entidad receptora envía un mensaje de tipo *DLL_SM_ERROR* a la otra entidad. Este mensaje contiene un parámetro que especifica el error (en este caso la longitud del mensaje es 1). En la tabla A.3.2 se muestran los tipos de error que se pueden generar.

Tabla A.4.2 Causas de error en la DLL

Código del error (binario)	Descripción
0000 0001	<i>Checksum</i> equivocado
0000 0010	Longitud del mensaje equivocado
0000 0011	Tipo de mensaje desconocido
0000 0100	Mecanismo de extensión no soportado
1111 1111	Causa de error desconocida

Si se genera un error en la transmisión del mensaje, la DLL de la entidad que envió el mensaje de texto, vuelve a enviar el mensaje. Si el mensaje vuelve a generar error, la conexión entre las dos entidades finalizará.

La repuesta del mensaje de la DLL es controlada por un temporizador que se activa al transmitir cada trama. El valor de este temporizador deberá ser: $T_{12} = 4000$ ms; si este tiempo expira (sin tener respuesta), la conexión se terminará.

Capa de Transferencia (TL)

La capa de transferencia provee el servicio a la capa de aplicación del mensaje (SM-AL). Permite el transporte de mensajes cortos de texto entre las entidades, recibiendo los mensajes desde una entidad y recibiendo los reportes de respuesta de estas transferencias. La longitud máxima de los mensajes de esta capa es de 176 octetos.

Existen 4 tipos de mensaje de la capa de transferencia (SM-TL), los cuales se describen a continuación:

Tabla A.4.3 Tipos de mensajes de capa de transferencia

Tipo de PDU	Descripción
SMS-DELIVER	Lleva un mensaje de texto (SM) desde el SC hasta el MS o al SMTE
SMS-DELIVER-REPORT	Lleva: a) una cláusula de fallo si es necesario b) información sobre una positiva o negativa respuesta de un SMS-DELIVER
SMS-SUBMIT	Lleva un mensaje de texto desde la SM-TE al SM-SC
SMS-SUBMIT-REPORT	Lleva: a) una cláusula de fallo si es necesario b) información sobre una positiva o negativa respuesta de un SMS-SUBMIT

Cada uno de estos mensajes están compuestos por PDU (unidad de descripción del protocolo), los cuales llevan diferentes tipos de información según sea el tipo de mensaje.

PDU del SM-TL

Para los parámetros dentro de los PDU hay tres formas de la representación numérica:

- Representación de Enteros
- Octetos
- Semi-octetos

Representación de Enteros: Pueden ser representados con un número completo de octetos o una fracción.

Representación de Octetos: El campo que está representado por un octeto, debe consistir siempre de un número completo de octetos. Cada octeto dentro de un campo representa un dígito decimal. El octeto con el número menor debe contener el más significativo dígito decimal.

Representación de semi-octetos: un campo que es la representación de un semi-octeto, debe consistir de un número completo de octetos, y quizás la mitad de un octeto. Cada mitad del octeto representa un dígito decimal.

Campos de Direcciones: Cada campo de dirección usado por el SM-TL consiste de los siguientes subcampos: un campo de la longitud de la dirección de un octeto (Address-Length), un campo para el tipo de dirección de un octeto (Type-of-Address), y un campo para el valor de la dirección de longitud variable (Address-Value), como se muestra abajo:

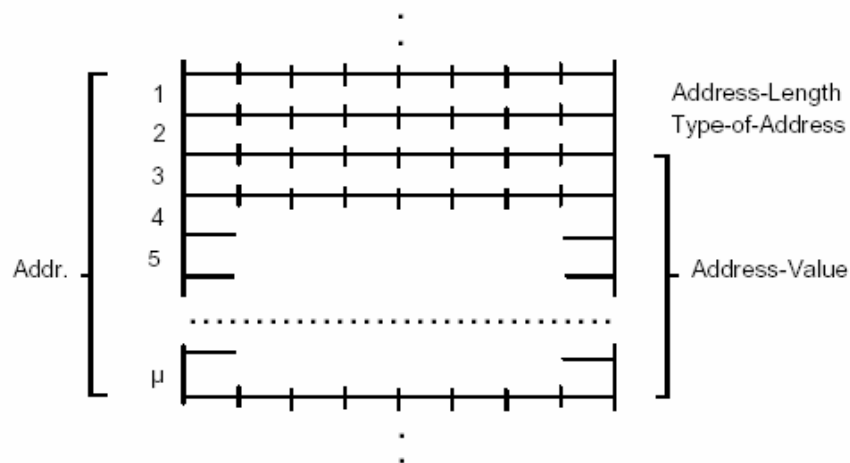


Figura A.4.3 Estructura de los campos de direcciones en el SM-TL¹

La longitud de la dirección es un entero que representa el número útil de semi-octetos dentro del campo *Address-Value*. El tipo de dirección es un octeto que representa el tipo de numeración del país. El valor de la dirección se da en semi-octetos y se debe realizar un intercambio de los mismos; por ejemplo, la representación de 0x30h debe guardarse como 0x03h.

A continuación se detalla la estructura de cada uno de los tipos de mensajes de la capa de transferencia.

¹ Tomado de ETSI TS 123 040

SMS-SUBMIT:

Este tipo de mensaje es enviado por la terminal del usuario que envía el mensaje de texto y es recibido por el centro de mensajes. En la figura A.4.4 se muestra la estructura del paquete.

TP-RP	TP-UDHI	TP-SRR	TP-VPF	TP-RD	TP-MTI
Referencia del mensaje (TP-MR) 1 entero					
Dirección de destino (TP-DA) 2 -12 octetos					
Identificador del protocolo (TP-PID) 1 octeto					
Esquema de codificación del dato (TP-DCS) 1 octeto					
Período de validación (TP-VP) 1 - 7 octetos					
Longitud del dato del usuario (TP-UDL) 1 entero					
Dato del usuario (TP-UD) Mensaje de texto					

Figura A.4.4 Estructura del paquete *SMS-SUBMIT*

SMS-SUBMIT-REPORT

Cuando el centro de mensajes recibe un *SMS-Submit*, éste deberá contestarle a la terminal del cliente origen (el que envía el mensaje de texto) con un *SMS-Submit-Report*. Esto sirve para informarle a la terminal del cliente que el mensaje de texto fue recibido por el centro de mensajes de forma correcta o incorrecta; por lo tanto, pueden haber dos tipos de respuesta: positiva o negativa. A en las figura A.4.5 y A.4.6 se muestra los elementos internos y la estructura del paquete para cada caso.

SMS-SUBMIT-REPORT (Respuesta negativa)

Este tipo de repuesta es enviada por el centro de mensajes cuando detecta algún tipo de error de capa 3 en el mensaje (*SMS-Submit*) recibido previamente.

TP-UDHI (1 bit)	TP-MTI (2 bits)
Causa de fallo (TP-FCS) 1 entero	
Indicador de parámetros (TP-PI) 1 octeto	
Tiempo de sellado (TP-SCTS) 7 octetos	
Identificador del protocolo (TP-PID) 1 octeto	
Esquema de codificación (TP-DCS) 1 octeto	
Longitud del dato del usuario (TP-UDL) 1 entero	
Dato del usuario (TP-UD)	

Figura A.4.5 Estructura del paquete *SMS-SUBMIT-REPORT* (negativo)

SMS-SUBMIT-REPORT (respuesta positiva)

Esta respuesta es enviada por el centro de mensajes cuando el mensaje de texto recibido anteriormente es correcto.

TP-UDHI (1 bit)	TP-MTI (2 bits)
Indicador de parámetros (TP-PI) 1 octeto	
Tiempo de sellado (TP-SCTS) 7 octetos	
Identificador del protocolo (TP-PID) 1 octeto	
Esquema de codificación (TP-DCS) 1 octeto	
Longitud del dato del usuario (TP-UDL) 1 entero	
Dato del usuario (TP-UD)	

Figura A.4.6 Estructura del paquete *SMS-SUBMIT-REPORT* (positivo)

SMS-DELIVER

Este tipo de mensaje es enviado por el centro de mensajes hacia la terminal del cliente destino. En este momento se realiza la entrega del mensaje de texto a su destinatario. La estructura del paquete se puede observar en la figura A.4.7.

TP-RP	TP-UDHI	TP-MMS	TP-MTI
Dirección original (TP-OA) 2 - 12 octetos			
Identificador del protocolo (TP-PID) 1 octeto			
Esquema de codificación (TP-DCS) 1 octeto			
Tiempo de sellado (TP-SCTS) 7 octetos			
Longitud del dato del usuario (TP-UDL) 1 entero			
Dato del usuario (UD) Mensaje de texto			

Figura A.4.7 Estructura del paquete *SMS-Deliver*

SMS-DELIVER-REPORT

Este tipo de mensaje es enviado por la terminal de destino hacia el centro de mensajes, como respuesta a un *SMS-Deliver*. Tiene como función avisar al centro de mensajes si el mensaje de texto se recibió de forma correcta o si se produjo un error en la capa 3. En las figuras A.4.8 y A.4.9 se puede observar las estructuras de los paquetes de ambos casos.

SMS-SUBMIT-REPORT (Respuesta negativa)

La respuesta negativa es enviada por la terminal cuando detecta algún tipo de error de capa 3 en el mensaje (*SMS-Deliver*) recibido previamente.

TP-UDHI (1 bit)	TP-MTI (2 bits)
Causa de fallo (TP-FCS) 1 entero	
Indicador de parámetros (TP-PI) 1 octeto	
Identificador del protocolo (TP-PID) 1 octeto	
Esquema de codificación (TP-DCS) 1 octeto	
Longitud del dato del usuario (TP-UDL) 1 entero	
Dato del usuario (TP-UD)	

Figura A.4.8 Estructura del paquete *SMS-SUBMIT-REPORT* (negativo)

SMS-SUBMIT-REPORT (respuesta positiva)

Esta repuesta es enviada por la terminal de destino cuando el mensaje de texto recibido anteriormente es correcto.

TP-UDHI (1 bit)	TP-MTI (2 bits)
Indicador de parámetros (TP-PI) 1 octeto	
Identificador del protocolo (TP-PID) 1 octeto	
Esquema de codificación (TP-DCS) 1 octeto	
Longitud del dato del usuario (TP-UDL) 1 entero	
Dato del usuario (TP-UD)	

Figura A.4.9 Estructura del paquete *SMS-SUBMIT-REPORT* (positivo)

Definición de los parámetros de los PDU

Indicador del tipo de mensaje (TP-MTI)

El indicador del tipo de mensaje son dos bits localizados en el primer octeto de todos los PDU, se ubica en los bits 0 y 1, estos pueden tomar los siguientes valores:

Tabla A.4.4 Codificación de los bits del TP-MTI

Bit 1	Bit 2	Tipo de mensaje
0	0	SMS-DELIVER (desde el SC hacia el SMTE o el MS)
0	0	SMS-DELIVER-REPORT (desde el SMTE o el MS hacia el SC)
0	1	SMS-SUBMIT (desde el SMTE o el MS hacia el SC)
0	1	SMS-SUBMIT-REPORT (desde el SC hacia el SMTE o el MS)
1	1	Reservado

Si una SM-TE recibe un TP-MTI con un valor de reservado, deberá procesar el mensaje como si fuera un *SMS-DELIVER*, pero guardará el mensaje exactamente como lo recibe.

Más mensaje por enviar (TP-MMS)

EL TP-MMS consiste de un bit, localizado en el bit número dos del primer octeto del SMS-DEVILER.

Tabla A.4.5 Codificación del bit del TP-MMS

Bit 2	Descripción
0	Más mensajes están esperando a la SMTE en el SC
1	No hay más mensajes esperando

Formato del periodo de validación (TP-VPF)

El TP-VPF son dos bits, localizado dentro de los bits 3 y 4 del primer octeto del SMS-SUBMIT, y se da por los siguientes valores:

Tabla A.4.6 Codificación de los bits del TP-VPF

Bit 1	Bit 2	Tipo de mensaje
0	0	TP-VP campo no presente
0	1	TP-VP campo presente (formato relativo)
1	0	TP-VP campo presente (formato reforzado)
1	1	TP-VP campo presente (formato absoluto)

Cualquier valor no soportado puede ser rechazado por el SM-SC, retornando el valor del “TP-VPF no soportado” en el TP-FCS del SMS-SUBMIT-REPORT.

Referencia del mensaje (TP-MR)

El TP-MR da en representación de un entero un número de referencia para un SMS-SUBMIT. La SM-TE incrementa en 1 en cada SMS-SUBMIT que está siendo sometido. El valor a ser usado en cada SMS-SUBMIT es obtenido al leer el último TP-MR usado, luego se incrementa este valor en 1. Este número de referencia puede poseer valores que van desde 0 hasta 255.

Dirección Original (TP-OA)

El campo de la dirección original está formado acorde a las reglas expuestas en el campo de direcciones.

Dirección Destino (TP-DA)

El campo de la dirección destino está formado acorde a las reglas expuestas en el campo de direcciones.

Identificador del Protocolo (TP-PID)

Este consiste de un octeto, y definen el tipo de tecnología que se está empleando. El SM-SC puede rechazar un mensaje con un valor del identificador de protocolo reservado o no soportado.

Esquema de la codificación de datos (TP-DCS)

Consiste de un octeto en el que se define el tipo de codificación que lleva el mensaje de texto. El tipo de codificación puede ser de 7 bits, 8 bits o 16 bits.

Estampado de tiempo del centro de servicio (TP-SCTS)

El TP-SCTS se da en semi-octetos, y representa el tiempo local de la siguiente forma:

	Año	Mes	Día	Hora	Minutos	Segundos	Zona
Dígitos (semi-octetos)	2	2	2	2	2	2	2

Figura A.4.10 Estructura del TP-SCTS

La zona de tiempo indica la diferencia, expresada en cuartos de horas, entre la hora local y el GMT (Greenwich Mean Time). En el primero de los dos semi-octetos, el primer bit (bit 3 del séptimo octeto del TP-SCST) representa el signo algebraico de sus diferencia (0 = positivo, 1 = negativo).

Periodo de validación (TP-VP)

Formato relativo

Comprende un octeto representado en un entero, dando la longitud de la validación del periodo, contando desde cuando el *SMS-SUBMIT* es recibido por el SC. La representación del tiempo es como sigue:

Tabla A.4.7 Parámetros del TP-VP (formato relativo).

Valor del TP-VP	Valor del periodo de validación
0 a 143	$(TP-VP + 1) \times 5$ minutos
144 a 167	12 horas + $((TP-VP - 143) + 30)$ minutos
168 a 196	$(TP-CP - 167) \times 1$ día
197 a 255	$(TP-CP - 196) \times 1$ semana

Formato absoluto

El TP-VP comprende 7 octetos, representados en semi-octetos que dan el tiempo absoluto de la determinación del periodo de validación. La representación del tiempo es idéntica a la del TP-SCTS

Formato reforzado

Está comprendido en 7 octetos. La presencia de todos los octetos son mandatorios. El primer octeto indica la forma como serán usados los otros 6. Cualquier bit reservado o no usado deberá ser colocado en cero. Las características del primer octeto se presentan a continuación:

- Bit 7 (bit de extensión): Colóquelo en 1 si el indicador de funcionalidad del TP-VP está siendo extendido a otro octeto. La colocación de un cero indica que no hay más indicadores de funcionalidad del TP-VP extendidos en los octetos que siguen.
- Bit 6 (un solo SM): Colocar un 1 si se requiere componer una prueba de entrega. El TP-VP, donde se presente, será aplicable a un solo SM.
- Bits 5, 4 y 3 son reservados.
- El formato de los bits 2, 1 y 0 se ilustran en la siguiente tabla.

Tabla A.4.8 Codificación de los bits 2, 1 y 0 del TP-VP (formato reforzado)

Bits (2, 1 y 0)	Especificación
000	Periodo de validación no especificado.
001	El periodo de validación es como se especifica para el caso de formato relativo.
010	El periodo de validación es relativo a la representación de un entero y los siguientes octetos del TP-VP contienen el valor en un rango de 0 a 255, representado de 0 a 255 segundos. El valor de cero es indefinido y reservado para futuros casos.
011	El periodo de validación es relativo y representado en semi-octetos. Los siguientes 3 octetos contienen el tiempo relativo en Horas, Minutos y Segundos.
100	Reservado.
101	Reservado.
110	Reservado.
111	Reservado.

Dirección destino (TP-RA)

El TP-RA indica la dirección del SME que fue el destino de un previo mensaje, siendo sujeto a un reporte de estado. El formato es acorde a las reglas del campo de direcciones.

Longitud del dato del usuario (TP-UDL)

Es la representación del número de octetos que contiene el TP-UD (mensaje de texto). Si el TP-UDL es cero, el TP-UD no estará presente.

Causas de error (TP-FCS)

Este campo está compuesto por un octeto y es usado para informar las razones por las cuales falló la transferencia.

Indicador de encabezado del dato del usuario (TP-UDHI)

El TP-UDHI está representado por un bit, éste se ubica en el primer octeto, en el bit número 6 de los siguientes PDU: *SMS-SUBMIT*, *SMS-SUBMIT-REPORT*, *SMS-DELIVER*, *SMS-DELIVER-REPORT*.

Sus valores pueden ser:

- Bit 6 = 0. TP-UD contiene solo un SM.
- Bit 6 = 1. El inicio del TP-UD contiene un encabezado juntado al mensaje corto.

Dato del usuario (TP-UD)

Este campo puede comprender sólo un mensaje de texto o también un adicional encabezado, esto depende del valor del TP-UDHI.

Cuando el TP-UDHI tiene un valor de 0 el TP-UD comprende solo el mensaje de texto, donde los datos pueden ser de 7 bits (alfabeto por defecto), de 8 bits o de 16 bits. Cuando el TP-UDHI tiene un valor de 1 el TP-UD contiene un encabezado en el primer octeto.

Rechazar duplicados (TP-RD)

El TP-RD constituye de un bit, localizado en el segundo bit del primer octeto del *SMS-SUBMIT*, el TP-RD tiene los siguientes valores:

- Bit 2 = 0. Instrucción del SC para aceptar un *SMS-SUBMIT* desde una SMTE, que todavía se mantiene dentro del SM-SC con el mismo TP-MR y el mismo TP-DA de un previo SM del mismo OA.
- Bit 2 = 1. Instrucción del SM-SC para rechazar un *SMS-SUBMIT* desde una SM-TE, que todavía se mantiene dentro del SC con el mismo TP-MR y el mismo TP-DA de un previo SM del mismo OA. En este caso un apropiado valor del TP-FCS será retornado en un *SMS-SUBMIT-REPORT*.

Identificador de parámetros (TP-PI)

El TP-PI comprende un número entre 1 octeto y 'n' octetos (dependiendo del bit de extensión), donde cada bit puesto en 1 indica que una opción particular de parámetro está presente en el campo que sigue. La estructura del TP-PI se muestra en la siguiente figura:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bit de extensión	Reservado	Reservado	Reservado	Reservado	TP-UDL	TP-DCS	TP-PID

Figura A.4.11 Estructura del TP-PI

El bit más significativo dentro del primer octeto y en cualquier otro octeto del TP-PI puede ser considerado como reservado para un bit de extensión, que cuando se coloca en 1 indica que otro octeto de TP-PI sigue inmediatamente después.

Si el bit del TP-UDL se coloca en cero, entonces por definición, ninguno de los campos del TP-UDL y el TP-UD pueden estar presentes. Si un bit reservado es colocado en 1, entonces la entidad que recibe deberá ignorar la colocación. La colocación de este bit significará que una información adicional seguirá el TP-UD, así que cualquier Terminal que reciba deberá descargar cualquier de los octetos que siguen después del TP-UD.